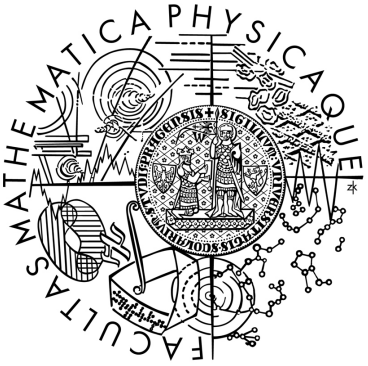
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Gabriela Znamenáčková

Vícestupňové úlohy stochastického programování a metoda scénářů

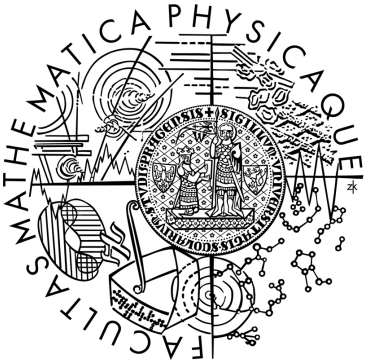
Katedra pravděpodobnosti a matematické statistiky

Vedoucí diplomové práce:	doc. RNDr. Petr Lachout, CSc.
Studijní program:	Matematika
Studijní obor:	Pravděpodobnost, matematická statistika a ekonometrie

Praha 2013

Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Gabriela Znamenáčková

Multistage stochastic programs and scenario approach

Department of Probability and Mathematical Statistics

Supervisor of the master thesis: doc. RNDr. Petr Lachout, CSc.
Study programme: Mathematics
Specialization: Probability, Mathematical Statistics and Econometrics

Prague 2013

I would like to express my sincere gratitude to my supervisor, doc. RNDr. Petr Lachout, CSc., for his guidance and valuable remarks in all the time of writing my master thesis.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague 5. 12. 2013 Gabriela Znamenáčková

Název práce: Vícestupňové úlohy stochastického programování a metoda scénářů

Autor: Gabriela Znamenáčková

Katedra: Katedra pravděpodobnosti a matematické statistiky, MFF UK

Vedoucí diplomové práce: doc. RNDr. Petr Lachout, CSc.

Abstrakt: Tato diplomová práce je zaměřena na úlohy stochastické optimalizace, ve kterých podstatně figurují náhodné vstupy, a navíc mají speciální, vícestupňovou strukturu. Přibližné řešení těchto složitých systémů umožňuje metoda scénářů. Existuje mnoho různých přístupů, jak scénáře a scénářové stromy generovat, v práci je přehled a popis hlavních z nich. Z pohledu numerické řešitelnosti jsou důležité také metody redukce scénářového stromu, jejichž hlavní myšlenky jsou taktéž zahrnuty. Mnoho rozhodovacích problémů z praxe má vícestupňovou strukturu s náhodnými vstupy. Lze se s nimi setkat např. v oblasti financí či v průmyslu, a tak je několik příkladů v závěru práce uvedeno jako ukázka propojení uvedené teorie s praxí.

Klíčová slova: Vícestupňové úlohy stochastického programování, metoda scénářů, generování scénářů

Title: Multistage stochastic programs and scenario approach

Author: Gabriela Znamenáčková

Department: Department of Probability and Mathematical Statistics, MFF UK

Supervisor: doc. RNDr. Petr Lachout, CSc.

Abstract: This diploma thesis focuses on stochastic optimization problems, in which uncertain inputs play the key role and additionally, they have a special multistage structure. A scenario approach enables us to get an approximate solution of these complicated problems. There are various methods of scenario and scenario tree generation, the main methods are described in the thesis. Scenario tree reduction methods are important from the numerical tractability point of view and the main ideas of scenario tree reduction are also included. Many practical decision problems have the multistage structure and they are affected by uncertainty. They are very common in the field of finance and industry, therefore at the end, some examples as illustration of the link between theory and practice are shown.

Keywords: Multistage stochastic programs, scenario method, generation of scenarios

Obsah

Introduction

Almost every system includes some level of uncertainty, which can play the central role in many decision problems. Uncertain parameters can't be simply ignored without losing accuracy and therefore making wrong decisions. There are several ways how to deal with uncertainty in decision problems. The main idea is to assign a probability distribution to the unknown parameters, supposing we have enough statistical data available to do that. If there are no statistical data available, the probability measure could only be based for example on a subjective experience of an expert in the branch, in which the decisions should be made (e.g. financial and production planning). In many cases, we are considering a system of future decisions, that should be made at specific times, and future observations of a random parameters (e.g. interest rates) between two future times. This type of decision process is called multistage stochastic program. The purpose of my diploma thesis is to introduce the concept of the multistage stochastic programs. As it was mentioned, we assign a probability distribution to the uncertain parameters, that enter the problem. The assumption, that the probability distribution of the random data, which enter the problem, is known, is very often unrealistic. Therefore, we try to approximate the true probability distribution in some way. A quite reasonable is the *scenario approach*, when we try to approximate the true probability distribution of the stochastic process by a discrete probability distribution carried on a finite number of atoms (*scenarios*). The stochastic data are often organized in the form of a *scenario tree* (e.g. for multistage stochastic linear programs).

There are many methods of scenario and scenario tree generation. A vast majority of them is mentioned in Chapter 6 and some of them are described into more details. With the increasing number of scenarios, the multistage stochastic program becomes very large. Consequently, this led to methods for reduction of the scenario tree (they are also mentioned in Chapter 6).

A lot of problems in reality have the multistage structure, so the theory of multistage stochastic programming and scenario approach can be used in many fields of human activity. There are several practical examples in Chapter 7 to illustrate the wide range of fields, where this theory can be used. In addition to that, it is shown, how to construct and reduce a scenario tree in GAMS\SCENRED 2.

1. Introduction to Stochastic Programming

In this chapter, the motivation example and the basic stochastic programming model will be stated. Subsequently, some basic concepts, definitions and relations from the probability and optimization theory useful for the development of stochastic programming models will be given.

1.1 Example of Motivation

As it was mentioned before, stochastic optimization deals with problems that arise from very common practical situations in financial management, electricity management and many other fields of human activity. A very illustrative example is the following problem inspired by [42]. Suppose you were running an exclusive restaurant that offers seafood specialties. Before the next week starts, you have to decide how many kilograms x of seafood you should purchase at price p for one kilogram. One kilogram of seafood dishes is sold at the average price s at your restaurant. Because the restaurant is luxurious and you take pride in offering only very fresh seafood, at the end of the week, you can sell the remained raw unsold seafood to another (not so renowned) restaurant at price r. The natural assumption is that $0 \leq r < p < s$. Let D denotes the number of kilograms of seafood sold in your restaurant per week (*demand*). If the amount of seafood sold in your restaurant per particular week is equal to the already bought amount x of seafood at the beginning of this particular week, you make a profit of $(s-p)x$. Otherwise your profit is equal to $(r-p)x + (s-r)D$. Your aim is to maximize the profit, which could be expressed as a function of x and D called the *objective function*, in this case as

$$f(x, D) = \begin{cases} (s-p)x, & x \leq D \\ (r-p)x + (s-r)D, & x > D. \end{cases}$$

If the demand D was known, the optimal amount of kilograms x^* would be the same as the demand D, $x^* = D$. But unfortunately, at the time we make a decision about how much seafood we should buy, the future demand D is not known. And this is the difficulty, which stochastic optimization can handle.

Suppose you have been collecting data about the demand for a long time. We can view the demand D as a *random variable*, whose probability distribution could be estimated by the cumulative distribution function (*cdf*) denoted by *G(y)*. The average profit over a very long period of time converges to the expected value by the Law of Large Numbers, so it makes a good sense to maximize the expected value of the objective function.

However, practically we have to take into account some other circumstances, e.g. the limited amount of money in the time of purchase of seafood, limited amount of seafood that could be sold to some other restaurant at the end of the week etc. It means that we aim at maximizing the expected value of the objective function subject to some constraints. In the next section, the basic model for approaching such problems will be discussed.

1.2 The stochastic programming model

In the last section, we tried to maximize the profit, that was modeled by the objective function $f(x, D)$, or its expected value. For some other problems, our target is to minimize the objective function (e.g. minimizing the cost of some process).

The general form introduced in [14] of the *mathematical program* in $\mathbb{R}^n$ could be written as:

$$\begin{aligned} & min\ f(x_1, \ldots, x_n) \\ subject\ to & \\ & h_k(\mathbf{x}) = 0,\ k = 1, \ldots, p \\ & g_l(\mathbf{x}) \leq 0,\ l = 1, \ldots, m \\ & \mathbf{x} \in \mathcal{X}_0. \end{aligned} \tag{1.2.1}$$

The constraints define the *set of feasible solutions* $\mathcal{X} = \{\mathbf{x} = (x_1, x_2, \ldots, x_n) : h_k(\mathbf{x}) = 0, k = 1, ..., p, g_l(\mathbf{x}) \leq 0, l = 1, ..., m, \mathbf{x} \in \mathcal{X}_0\}$, where f is a real function and also h_k for $\forall\, k$ and g_k for $\forall\, l$ are real functions, $\mathcal{X}_0$ is a set of specific conditions for $\mathbf{x}$ (e.g. integrability). All of the previously mentioned functions may depend on parameters (*parameter programs*). In case of random parameters, we talk about *stochastic programs*.

Let the objective function and the constraint functions be dependent on a random parameter ω [1], which is as an element of a sample space Ω equipped with a σ-algebra $\mathcal{F}$, and its probability distribution is P. Let (Ω, $\mathcal{F}$, P) be a probability space. According to the different realizations of the random parameter ω, we would have different optimal values for the stochastic programming problem and we would not be able to decide which solution is the "best". As it was suggested in the previous example, a possible way of solving that problem is to maximize the expected value of the objective function. We will aim at solving the following problem:

$$min\ f_E(\mathbf{x}, \omega) := \mathbf{E}[f(\mathbf{x}, \omega)], \mathbf{x} \in \mathcal{X}. \tag{1.2.2}$$

To find a solution for the problem as it is formulated above, we assume the knowledge of the probability distribution P on (Ω, $\mathcal{F}$) and that the expected value is well defined. Function $f_E(\mathbf{x}, \omega)$ is called *expectation function*.

We also need to adjust the constraints to the case of random parameters, since for some values of ω it is impossible to meet the required conditions and it is unrealistic to want that that constraints has to hold for all realizations of ω. For dealing with this issue, the same approach of the expected value as for the objective function could be used. We will then have the constraint functions $h_k^E(\mathbf{x}, \omega) := \mathbf{E}[h_k(\mathbf{x}, \omega)],\ k = 1, \ldots, p$ and $g_l^E(\mathbf{x}, \omega) := \mathbf{E}[g_l(\mathbf{x}, \omega)],\ l = 1, \ldots, m$.

[1] The random parameter ω can be a random variable or a random vector. In the whole thesis, generally it will not be written as bold symbol (usually used for vectors) to distinguish these both cases. It will be written in bold only in case, when we want to emphasize, that it is a vector.

Another approach could be to use the *probabilistic* or *chance* constraints for the functions g_l, which means that we require, that the conditions has to be met with a certain probability, $P[g_l(\mathbf{x}, \omega) \leq 0] \geq 1 - \alpha,\ l = 1, \ldots, m$, where $\alpha \in (0, 1)$ is fixed. If the probability distribution of $h_k(\mathbf{x}, \omega)$ is discrete, we can also use the probabilistic constraints to them.

1.3 Basic concepts and definitions

In the whole thesis, basic knowledge of the probability theory is expected. However, some terms and relationships especially needed for the optimization will be briefly mentioned. Theorems and propositions are usually given without proofs. The source of section 1.3 is [42].

1.3.1 Expectation functions

Let $\bar{\mathbb{R}}$ denote the extended real numbers, $\bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$ and Z be a random variable. Optimal values of the optimization problems can also take values $+\infty$ and $-\infty$, therefore the random variables that occur in our problems should be also allowed to take these values, which means that Z is a measurable *extended function*, $Z : \Omega \to \bar{\mathbb{R}}$, called *extended random variable*. The *expected value* of an extended random variable Z is defined as:

$$\mathbf{E}_P[Z] := \int_{\Omega} Z(\omega)\, dP(\omega).$$

If there is no doubt about what probability measure is taken into account, we can simply write $\mathbf{E}[Z]$. Let $Z^+ := max\{0, Z\}$, we say that the expected value $\mathbf{E}[Z] = \mathbf{E}[Z^+] - \mathbf{E}[(-Z^+)]$ is *well-defined*, if it does not come both $\mathbf{E}[Z^+]$ and $\mathbf{E}[(-Z^+)]$ to be $+\infty$. If the expected value $\mathbf{E}_P[Z]$ is well defined and finite, than we say that the random variable Z is *P-integrable*.

The objective function $f(\mathbf{x}, \omega)$ for a given $\mathbf{x}$ can be treated as a random variable. The the expectation function $f_E(\mathbf{x}, \omega) := \mathbf{E}[f(\mathbf{x}, \omega)]$ is *well-defined*, if for every $\mathbf{x} \in \mathcal{X}$ the objective function $f(\mathbf{x}, \cdot)$ is measurable and as well either $\mathbf{E}[f(\mathbf{x}, \omega)^+] < +\infty$ or $\mathbf{E}[(-f(\mathbf{x}, \omega))^+] < +\infty$.

The *domain* of a function f is defined as:

$$\operatorname{dom} f := \{\mathbf{x} \in \mathbb{R}^n : f(x) < +\infty\}.$$

The effective feasible set is then given by $\mathcal{X} \cap \operatorname{dom} f$.

We say that the extended real valued function f is *proper*, if for all $\mathbf{x} \in \mathbb{R}^n$ holds $f(\mathbf{x}) > -\infty$ and its domain is nonempty.

Some of the properties of the objective function $f(\mathbf{x}, \omega)$ are inherited by the expected value function $f_E(\mathbf{x}, \omega) := \mathbf{E}[f(\mathbf{x}, \omega)]$. For a fixed $\mathbf{x} \in \mathbb{R}^n$, if the objective function $f(\cdot, \omega)$ is convex for P- almost every $\omega \in \Omega$, then the expected value function $f_E(\cdot, \omega)$ is also convex.

There is a possibility to modify the objective function involving the constraints. We can consider the following objective function for any $\omega \in \Omega$, whose

expected value will be minimized:

$$\bar{f}(\mathbf{x},\omega) := \begin{cases} f(\mathbf{x},\omega), & \mathbf{x} \in \mathcal{X} \\ +\infty, & \mathbf{x} \notin \mathcal{X}. \end{cases}$$

Then we can rewrite our task (1.2.2) as

$$\min \mathbf{E}[\bar{f}(\mathbf{x},\omega)],\ \mathbf{x} \in \mathbb{R}^n\ ^2.$$

1.3.2 Lower semicontinuous functions

Definition 1. The extended real valued function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is *lower semicontinuous* (abbreviation *lsc*)*at a certain point* $\mathbf{x}_0 \in \mathbb{R}^n$, if

$$\liminf_{\mathbf{x}\to\mathbf{x}_0} f(\mathbf{x}) \geq f(\mathbf{x}_0).$$

We say that function f is *lower semicontinuous*, if it is *lsc* at every point $\mathbf{x} \in \mathbb{R}^n$.

It could be shown, that if the *epigraph* of the function f, $\operatorname{epi} f := \{(\mathbf{x},\beta) : f(\mathbf{x}) \leq \beta\}$, is a closed subset of $\mathbb{R}^n \times \mathbb{R}$, function f is *lsc*.

Definition 2. We call an extended real valued function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ *polyhedral*, if it is proper convex, lower semicontinuous, its domain is a convex closed polyhedron and it is piecewise linear on its domain.

Theorem 1. *Suppose that for function* $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ *holds* $-\infty < f(\mathbf{x})$ *for every* $\mathbf{x} \in \mathbb{R}^n$. *Let function* f *is* lsc *and* $\operatorname{dom} f$ *is bounded. Then the set of all optimal solutions* $\arg\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ *is nonempty.*

In the next proposition, the relationship between the lower semicontinuity of the objective function $f(\mathbf{x},\omega)$ and the lower semicontinuity of the expected value function $f_E(\mathbf{x},\omega)$ will be outlined.

Proposition 2. *Suppose that the following conditions are met:*

(i) *the objective function* $f(\cdot,\omega)$ *is* lsc *at* $\boldsymbol{x}_0 \in \mathbb{R}^n$ *for* P*-almost every* $\omega \in \Omega$,

(ii) *for every* $\boldsymbol{x} \in \mathbb{R}^n$ *in the neighborhood of* $\boldsymbol{x}_0$ *the objective function* $f(\cdot,\omega)$ *is measurable*,

(iii) *there exists a* P*-integrable function* $Z(\omega)$ *such that for* P*-almost all* $\omega \in \Omega$ *and for all* $\boldsymbol{x}$ *in a neighborhood of* $\boldsymbol{x}_0$ *holds* $f(\boldsymbol{x},\omega) \geq Z(\omega)$.

Then the expected value function $f_E(\boldsymbol{x},\omega) := \boldsymbol{E}[f(\boldsymbol{x},\omega)]$ *is well defined for all* $\boldsymbol{x}$ *in the neighborhood of* $\boldsymbol{x}_0$ *and* lsc *at* $\boldsymbol{x}_0$.

Proof. Expected value function $f_E(\mathbf{x},\omega)$ is well defined for $\mathbf{x} \in \mathbb{R}^n$ in a neighborhood of $\mathbf{x}_0$ (follows from (ii) and (iii)). Assuming (iii) and using Fatou's lemma,

[2] In general, $\mathbf{x}$ is an element of the space of decision variables $\mathcal{Y}$, but we often identify it with $\mathbb{R}^n$.

we have that

$$\liminf_{\mathbf{x}\to\mathbf{x}_0} \int_\Omega f(\mathbf{x},\omega)\,dP(\omega) \geq \int_\Omega \liminf_{\mathbf{x}\to\mathbf{x}_0} f(\mathbf{x},\omega)\,dP(\omega)$$

Adding the assumption (i) we obtain that $f(\mathbf{x},\omega)$ is *lsc* at $\mathbf{x}_0$.

Q. E. D.

The probabilistic constraints $P[g_l(\mathbf{x},\omega) \leq 0] \geq 1-\alpha,\ l = 1,\dots,m$ can be written in the form $\mathbf{E}[\mathbf{1}_{(0,+\infty)}g_l(\mathbf{x},\omega)] \leq \alpha,\ l = 1,\dots,m$. Assume that the functions $g_l(\cdot,\omega)$ are *lsc* for P-almost every $\omega \in \Omega$ and functions $g_l(\mathbf{x},\cdot)$ are measurable for all $\mathbf{x} \in \mathbb{R}^n$. Functions $\mathbf{1}_{(0,+\infty)}g_l(\cdot,\omega)$ are then also *lsc* for P-almost every $\omega \in \Omega$ and bounded. Using the Proposition 2 we have that the expected value functions $\mathbf{E}[\mathbf{1}_{(0,+\infty)}g_l(\mathbf{x},\omega)]$ are *lsc*. Therefore the probabilistic constraints define a closed subset of $\mathbb{R}^n$.

1.3.3 Optimal values and solutions

In the optimization problems as they were defined, our aim was to minimize the objective function subject to some constraints. However, in many practical problems we would like to maximize the objective function. We will consider a function $h : \mathbb{R}^n \times \mathbb{R}^m \to \bar{\mathbb{R}}$ and the corresponding functions $\phi(\mathbf{x}) := \inf_{\mathbf{y}\in\mathbb{R}^m} h(\mathbf{x},\mathbf{y})$ and $\psi(\mathbf{x}) := \sup_{\mathbf{y}\in\mathbb{R}^m} h(\mathbf{x},\mathbf{y})$.

Proposition 3. *Suppose that functions $\phi(\boldsymbol{x})$ and $\psi(\boldsymbol{x})$ are defined as it was stated above. Then the following holds*:

(i) *if for every $\boldsymbol{y} \in \mathbb{R}^m$ the function $h(\cdot,\boldsymbol{y})$ is* lsc, *then the function $\psi(\boldsymbol{x})$ is* lsc,

(ii) *the function $h(\cdot,\cdot)$ is* lsc *and if there exists such a bounded set $S \subset \mathbb{R}^m$, that* $\operatorname{dom} h(\mathbf{x},\cdot) \subset S$ *for all $\boldsymbol{x} \in \mathbb{R}^n$, then the function $\phi(\boldsymbol{x})$ is* lsc.

Definition 3. A mapping $\mathcal{G}$ that assigns to each $\omega \in \Omega$ a subset of $\mathbb{R}^n$ is called a *multifunction* $\mathcal{G} : \Omega \to \mathbb{R}^n$.
If for every $\omega \in \Omega$ is $\mathcal{G}(\omega)$ a closed subset of $\mathbb{R}^n$, we say that $\mathcal{G}$ is *closed valued*.
We say that a closed valued multifunction is *measurable*, if the inverse image $\mathcal{G}^{-1}(A) := \{\omega \in \Omega : \mathcal{G}(\omega) \cap A \neq \emptyset\}$ is $\mathcal{F}$-measurable for every closed set $A \subset \mathbb{R}^n$.

If $\mathcal{G}$ is measurable, the domain of $\mathcal{G}$ is $\operatorname{dom}\mathcal{G} := \{\omega \in \Omega : \mathcal{G}(\omega) \neq \emptyset\} = \mathcal{G}^{-1}(\mathbb{R}^n)$ is an $\mathcal{F}$-measurable subset of Ω.

Definition 4. We call a mapping $G : \operatorname{dom}\mathcal{G} \to \mathbb{R}^n$ a *selection* of $\mathcal{G}$, if for all $\omega \in \operatorname{dom}\mathcal{G}$ holds $G(\omega) \in \mathcal{G}(\omega)$. We say, that the mapping G is a *measurable selection* of $\mathcal{G}$, if G is measurable.

Theorem 4. Castaining Representation theorem. *Let $\mathcal{G} : \Omega \to \mathbb{R}^n$ is a closed valued multifunction. Suppose that:*

(i) *dom$\mathcal{G}$ is an $\mathcal{F}$-measurable subset of Ω ,*

(ii) *there exists such a countable family $\{G_i, i \in \mathcal{N}\}$ of measurable selections of $\mathcal{G}$, so that for every $\omega \in \Omega$ the set $\{G_i(\omega), i \in \mathcal{N}\}$ is dense in $\mathcal{G}(\omega)$.*

Then $\mathcal{G}$ is measurable.

Definition 5. We say that function $(\mathbf{x}, \omega) \to f(\mathbf{x}, \omega)$ is *random lower semicontinuous*, if the corresponding epigraphical multifunction $\omega \to \text{epi } f(\cdot, \omega)$ is closed valued and measurable.

Because of the close valuedness of the epigraphical multifunction, the epigraph epi$f(\cdot, \omega)$ is a closed subset of $\mathbb{R}^n \times \mathbb{R}$, thus $f(\cdot, \omega)$ is *lsc.*

Theorem 5. *Assuming that σ-algebra $\mathcal{F}$ is P-complete and the following conditions hold:*

(i) *function $f(\cdot, \omega)$ is* lsc *for every $\omega \in \Omega$,*

(ii) *function $f(\cdot, \cdot)$ is measurable with respect to the σ-algebra of $\mathbb{R}^n \times \Omega$ given by the product of σ-algebras $\mathcal{B}$ and $\mathcal{F}$.*

Then the extended real valued function $f : \mathbb{R}^n \times \Omega \to \bar{\mathbb{R}}$ is random lsc.

We will consider the following optimal value function

$$\vartheta(\omega) := \inf_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, \omega)$$

and the respective set of optimal solutions

$$\mathcal{X}^*(\omega) := \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, \omega).$$

Theorem 6. *Suppose that $f : \mathbb{R}^n \times \Omega \to \bar{\mathbb{R}}$ is a random* lsc *function. Then both the optimal value function $\vartheta(\omega)$ and the optimal solution multifunction $\mathcal{X}^*(\omega)$ are measurable.*

Theorem 7. *Let consider a random* lsc *function $f : \mathbb{R}^{n+m} \times \Omega \to \bar{\mathbb{R}}$ and the corresponding optimal value function $\vartheta(\boldsymbol{x}, \omega) := \inf_{\boldsymbol{y} \in \mathbb{R}^m} f(\boldsymbol{x}, \boldsymbol{y}, \omega)$. Suppose that there exists such a bounded set $S \subset \mathbb{R}^m$, that for all $(\boldsymbol{x}, \omega) \in \mathbb{R}^n \times \Omega$ the* dom $f(\mathbf{x}, \cdot, \omega) \subset S$.*Then the optimal value function $\vartheta(\boldsymbol{x}, \omega)$ is* lsc.

1.3.4 Conjugate Function and Subdifferentials

In this subsection, a brief introduction to the theory of subdifferentials and conjugate functions will be made using [45].

Definition 6. We call a linear space $\mathcal{W}$ equipped with a norm $\|\cdot\|$ a *Banach space*, if it is complete (every Cauchy sequence in $\mathcal{W}$ has a limit).

The space of all linear continuous functionals $\zeta : \mathcal{W} \to \mathbb{R}$ forms the dual space of $\mathcal{W}$, which is denoted by $\mathcal{W}^*$. We denote by $\langle \zeta, z \rangle := \zeta(z)$ a scalar product on $\mathcal{W}^* \times \mathcal{W}$ for $\zeta \in \mathcal{W}^*$ and $z \in \mathcal{W}$.

Definition 7. Let $\mathcal{W}$ be a Banach space, $\mathcal{W}^*$ its dual space, $f : \mathcal{W} \to \bar{\mathbb{R}}$ an extended valued function. The *conjugate function* of f is defined as

$$f^*(\zeta) := \sup_{\zeta \in \mathcal{W}} \{\langle \zeta, z \rangle - f(z)\}.$$

The conjugate function $f^* : \mathcal{W}^* \to \bar{\mathcal{R}}$ is always convex and lsc.

Definition 8. A vector $\mathbf{t} \in \mathbb{R}^n$ is called a subgradient of $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ at $\mathbf{x}_0$ if

$$f(\mathbf{x}) - f(\mathbf{x}_0) \geq \mathbf{t}^T(\mathbf{x} - \mathbf{x}_0), \qquad \forall\, \mathbf{x} \in \mathbb{R}^n.$$

We call the set of all subgradients of $f(\mathbf{x})$ at $\mathbf{x}_0$ the *subdifferential* and denote it by $\partial f(\mathbf{x}_0)$. Function f is *subdifferentiable* at $\mathbf{x}_0$, if the set $\partial f(\mathbf{x}_0)$ is nonempty.

The subdifferential $\partial f(\mathbf{x}_0)$ is a closed convex subset of $\mathbb{R}^n$. If f is subdifferentiable at $\mathbf{x}_0$, then $f(\mathbf{x}) > -\infty$ for any $\mathbf{x}$ and therefore function f is proper.

2. Two-stage Problems

2.1 Linear Two-stage Problem

The example of motivation (1.1) in the first section is a good example of a *two-stage problem*. Before we know, what the demand D (number of kilograms of seafood) will be the next week, we need to make a decision about how many kilograms x of seafood to buy - *first stage decision variable*. At the end of the week, when the demand D is known, we will sell as many kilograms of the remaining seafood as possible. At the second stage, our *second stage decision variables* are the number of kilograms y that we sell at price s to our guests, and the quantity z that we resell to another (not very renowned) restaurant at price r. Given the amount of seafood bought x and the realization of the demand D, at the second stage we are aiming at maximizing the profit. The second stage problem can be expressed as:

$$\begin{aligned} & \max_{y,\,z} sy + rz \\ \textit{subject to} \quad & \\ & y \leq D, y + z \leq x, \\ & y \geq 0, z \geq 0 \end{aligned}$$

The optimal solution of the above stated second stage problem is $y^* = \min\{x, D\}$ and $z^* = \max\{x - D, 0\}$. The optimal value is the profit $f(x, D)$.

We can write a *two-stage stochastic linear program* in the form:

$$\begin{aligned} & \min_{\mathbf{x}\in\mathbb{R}^n} \mathbf{c}^T\mathbf{x} + \mathbf{E}[Q(\mathbf{x}, \boldsymbol{\xi}(\omega))] \\ \textit{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq 0, \end{aligned} \tag{2.1.1}$$

where $Q(\mathbf{x}, \boldsymbol{\xi}(\omega))$ is the optimal value of the second stage problem stated as follows)

$$\begin{aligned} & \min_{\mathbf{y}\in\mathbb{R}^m} \mathbf{q}^T(\omega)\mathbf{y} \\ \textit{subject to} \quad & \mathbf{T}(\omega)\mathbf{x} + \mathbf{W}(\omega)\mathbf{y} = \mathbf{h}(\omega), \quad \mathbf{y} \geq 0. \end{aligned} \tag{2.1.2}$$

Vectors $\mathbf{x}$ and $\mathbf{y}$ refer to the first and second stage decision variables, respectively. We will denote the set of feasible solutions of the first stage by $\mathcal{X} = \{\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{R}^n_+\}$. The data in the second stage problem are $\boldsymbol{\xi}(\omega) := (\mathbf{q}(\omega), \mathbf{h}(\omega), \mathbf{T}(\omega), \mathbf{W}(\omega))$ and we can regard all the elements of the vector $\boldsymbol{\xi}(\omega)$ as random. It can happen that only some of the elements are random. From now onwards we will write shorter $\boldsymbol{\xi} := (\mathbf{q}, \mathbf{h}, \mathbf{T}, \mathbf{W})$. The expectation operator at the first-stage problem (2.1.1) is taken with respect to the probability distribution of the random vector $\boldsymbol{\xi}$[3]. We suppose the probability distribution of $\boldsymbol{\xi}$ to be known. The support of

[3] The notation $\boldsymbol{\xi}$ will be used to denote both the random vector and its particular realization. If it is doubtful, then it will be written as $\boldsymbol{\xi} = \boldsymbol{\xi}(\omega)$.

the probability distribution of $\boldsymbol{\xi}$ will be denoted by $\Xi \subset \mathbb{R}^d$. If the second-stage problem (2.1.2) is infeasible for some $\mathbf{x}$ and $\boldsymbol{\xi} \in \Xi$, the function $Q(\mathbf{x}, \boldsymbol{\xi})$ takes the value $+\infty$. If the second-stage problem is unbounded from below, then $Q(\mathbf{x}, \boldsymbol{\xi})$ takes the value $-\infty$. It should be verified that the expected value is well defined. We call the matrix $\mathbf{T}$ *technology matrix* and the matrix $\mathbf{W}$ *recourse matrix*. According to the attributes of the recourse matrix $\mathbf{W}$, we can classify the stochastic linear program (SLP) as in is mentioned in [10] as follows:

- SLP with *fixed recourse*, if the recourse matrix $\mathbf{W}$ is fixed
- SLP with *complete fixed recourse*, if the recourse matrix $\mathbf{W}$ is fixed and if the system $\mathbf{Wy} = \mathbf{z}$ has a nonnegative solution for any right hand side $\mathbf{z}$
- SLP with *simple recourse* is a special case of the complete fixed recourse problems. We distinguish and penalize only two types of discrepancies y_i^+, y_i^- in rows of the system $\mathbf{Tx} + \mathbf{Wy} = \mathbf{h}$. Therefore the corresponding recourse matrix takes on the form $\mathbf{W} = (\mathbf{I}, -\mathbf{I})$, where $\mathbf{I}$ is the unit matrix of the matching dimension.
- SLP with *relatively complete recourse*, if the second stage problem (2.1.2) is feasible a.s. for an arbitrary $\mathbf{x} \in \mathcal{X}$ and $\omega \in \Omega$

In case of an infeasible second-stage problem for a first stage decision $\mathbf{x} \in \mathcal{X}$ and a realization of $\boldsymbol{\xi}$, accepting such decision may have disastrous consequences and infinite value of $Q(\mathbf{x}, \boldsymbol{\xi})$ is often used to reflect this situation. Another way of dealing with this situation is to adjust the constraints of the first-stage problem (2.1.1), so that we exclude first-stage decisions, that will lead to an infeasible second-stage problem. The set described by so-called *induced constraints* is denoted by $\mathcal{X}_I$. Then the first-stage problem is written as:

$$\min \mathbf{c}^T\mathbf{x} + \mathbf{E}[Q(\mathbf{x}, \boldsymbol{\xi})] \qquad \text{on the set } \mathcal{X} \cap \mathcal{X}_I \tag{2.1.3}$$

It can be proven that (2.1.3) is a convex program supposing that the recourse matrix $\mathbf{W}$ is fixed. There are another conditions that should be met to ensure that the objective function is well defined (e.g. a sufficient condition is the existence of all second order moments of the vector of all random parameters).
The set of induced constraints $\mathcal{X}_I$ for the special case of SLP with fixed recourse can be written as:

$$\mathcal{X}_I = \{\mathbf{x} : \exists\, \mathbf{y} \geq 0 \quad \text{such that} \quad \mathbf{Wy} = \mathbf{h} - \mathbf{Tx} \quad \text{a.s.}\} \tag{2.1.4}$$

The problem (2.1.3) is then equivalent with

$$\begin{aligned} &\min \mathbf{E}[\mathbf{c}^T\mathbf{x} + \mathbf{q}^T\mathbf{y}] \\ &\textit{subject to} \quad \mathbf{x} \in \mathcal{X} \quad \text{and} \quad \mathbf{y} \geq 0 \quad \text{such that} \\ &\qquad\qquad \mathbf{Wy} + \mathbf{Tx} = \mathbf{h} \qquad a.s. \end{aligned} \tag{2.1.5}$$

In the next subsection, the formulation of the linear two-stage problem for the case of a discrete distribution of $\boldsymbol{\xi}$ will be introduced.

2.1.1 Discrete Distributions

This subsection is inspired by [45] and [42]. In order to simplify the presentation we will suppose that the probability distribution of the random vector $\boldsymbol{\xi}$ has a finite support. That means that $\boldsymbol{\xi}$ has a finite number K of possible realizations $\boldsymbol{\xi}_k = (\mathbf{q}_k, \mathbf{h}_k, \mathbf{T}_k, \mathbf{W}_k)$ with corresponding (positive) probabilities p_k, $k = 1, \ldots, K$. The support in this case is therefore $\boldsymbol{\Xi} = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_K\}$. In that special case $\mathbf{E}[Q(\mathbf{x}, \boldsymbol{\xi})] = \sum_{k=1}^{K} p_k Q(\mathbf{x}, \boldsymbol{\xi}_k)$, where

$$Q(\mathbf{x}, \boldsymbol{\xi}_k) = \inf\{\mathbf{q}_k^T \mathbf{y}_k : \mathbf{T}_k \mathbf{x} + \mathbf{W}_k \mathbf{y}_k = \mathbf{h}_k, \mathbf{y}_k \geq 0\}. \tag{2.1.1.1}$$

The expectation $\mathbf{E}[Q(\mathbf{x}, \boldsymbol{\xi})]$ is equal to the optimal value of the following linear programming problem for a given $\mathbf{x}$:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_K} \quad & \sum_{k=1}^{K} p_k \mathbf{q}_k^T \mathbf{y}_k \\ \textit{subject to} \quad & \mathbf{T}_k \mathbf{x} + \mathbf{W}_k \mathbf{y}_k = \mathbf{h}_k, \\ & \mathbf{y}_k \geq 0, \quad k = 1, \ldots, K. \end{aligned} \tag{2.1.1.2}$$

If for at least one $k \in \{1, \ldots, K\}$ the corresponding second-stage program is infeasible, then the problem (2.1.1.2) is infeasible and therefore its optimal value is $+\infty$. The sum $\sum_{k=1}^{K} p_k Q(\mathbf{x}, \boldsymbol{\xi}_k)$ equals $+\infty$ if at least one of $Q(\mathbf{x}, \boldsymbol{\xi}_k) = +\infty$. We will assume here that $+\infty + (-\infty) = +\infty$.

The two-stage stochastic linear program (2.1.1) and (2.1.2) can be written in the form:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_K} \quad & \mathbf{c}^T \mathbf{x} + \sum_{k=1}^{K} p_k \mathbf{q}_k^T \mathbf{y}_k \\ \textit{subject to} \quad & \mathbf{T}_k \mathbf{x} + \mathbf{W}_k \mathbf{y}_k = \mathbf{h}_k, \qquad k = 1, \ldots, K, \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0, \quad \mathbf{y}_k \geq 0, \qquad k = 1, \ldots, K \end{aligned} \tag{2.1.1.3}$$

As it could be seen from the above stated formulation, the two-stage problem can be formulated as one large-scale linear programming problem. The size of the program (2.1.1.3) can be very large. As it is shown in [10], considering right hand sides $\mathbf{h}$ consisting of n independent random components with probability distributions approximated by alternative ones, for m-dimensional $\mathbf{b}$ we will obtain $m + 2^n$ equations as constraints.

2.1.2 Scenario Approach and Nonanticipativity Constraints

We will still assume the finite number of scenarios as in the previous subsection. The problem (2.1.1.3) can be relaxed by replacing vector $\mathbf{x}$ by K vectors, $\mathbf{x}_1, \ldots, \mathbf{x}_K$, one vector for each scenario. As in [42], this problem is formulated as:

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_K,\mathbf{y}_1,\ldots,\mathbf{y}_K} \sum_{k=1}^{K} p_k(\mathbf{c}^T\mathbf{x} + \mathbf{q}_k^T\mathbf{y}_k)$$

$$\begin{aligned} \textit{subject to} \qquad & \mathbf{T}_k\mathbf{x}_k + \mathbf{W}_k\mathbf{y}_k = \mathbf{h}_k \\ & \mathbf{A}\mathbf{x}_k = \mathbf{b} \\ & \mathbf{x}_k \geq 0, \quad \mathbf{y}_k \geq 0, \qquad k = 1,\ldots,K \end{aligned} \tag{2.1.2.1}$$

Problem (2.1.2.1) can be split into K smaller problems corresponding to each scenario and thus this is easier for numerical solution. However, formulating the problem like this, allows the first-stage decisions $\mathbf{x}_1,\ldots,\mathbf{x}_K$ to depend on the future realization of he random data of the second-stage. Adding additional constraints, so-called *nonanticipativity constraints*,

$$\mathbf{x}_k = \mathbf{x}_j, \qquad \text{for all} \quad 1 \leq k < j \leq K,$$

enables us to fix this and therefore problem (2.1.2.1) becomes equivalent to (2.1.1.3). Nonanaticipativity constraints guarantee that the first-stage decision variables do not depend on the realization of the random data. These constraints are especially important in the multistage stochastic programs, which will be discussed later on. It is possible to rewrite the nonanticipativity constraints in the form $\mathbf{x}_k = \mathbf{x}_{k+1}$ for $k = 1,\ldots,K-1$. Another way of writing the nonanticipativity condition (especially useful for the general formulation, which will be discussed later) is

$$\mathbf{x}_k = \sum_{i=1}^{K} p_i\mathbf{x}_i, \quad k = 1,\ldots,K.$$

2.1.3 Dual Problem and Optimality Conditions

The source for this section is [45]. The second-stage problem (2.1.2) is a linear programming problem and we can write its dual problem in the form:

$$\max_{\mathbf{d}} \mathbf{d}^T(\mathbf{h} - \mathbf{T}\mathbf{x})$$

$$\textit{subject to} \qquad \mathbf{W}^T\mathbf{d} \leq \mathbf{q} \tag{2.1.3.1}$$

Optimal values of the above stated dual problem (2.1.3.1) and the second-stage problem (2.1.2) are equal to each other, if none of these problems is infeasible. If the optimal values of both problems are finite, then the set of optimal solutions of each of these two problems is nonempty. We will consider the following function:

$$s_q(\mathbf{z}) := \inf\{\mathbf{q}^T\mathbf{y} : \mathbf{W}\mathbf{y} = \mathbf{z},\ \mathbf{y} \geq 0\}. \tag{2.1.3.2}$$

According to the definition of the function $Q(\mathbf{x}, \boldsymbol{\xi})$, it is clear that using (2.1.3.2) we have $Q(\mathbf{x}, \boldsymbol{\xi}) = s_q(\mathbf{h} - \mathbf{T}\mathbf{x})$. Let a set

$$\Pi(\mathbf{q}) := \{\mathbf{d} : \mathbf{W}^T\mathbf{d} \leq \mathbf{q}\}. \tag{2.1.3.3}$$

be considered. In case the set $\Pi(\mathbf{q})$ is nonempty, then

$$s_q(\mathbf{z}) := \sup_{\mathbf{d}\in\Pi(\mathbf{q})} \mathbf{d}^T\mathbf{z}, \tag{2.1.3.4}$$

$s_q(\cdot)$ is the support function of the set $\Pi(\mathbf{q})$, which is convex, closed and polyhedral, therefore it has a finite number of extreme points. Function $s_q(\cdot)$ is convex. When the set $\Pi(\mathbf{q})$ is empty, then $s_q(\mathbf{z})$ can take only the values $+\infty$ or $-\infty$. In case of a nonempty set $\Pi(\mathbf{q})$, the function $s_q(\cdot)$ is positively homogeneous polyhedral. Some propositions are stated below, proofs can be found in [45], p. 28-31.

Proposition 8. *The function $Q(\cdot,\boldsymbol{\xi})$ is convex for any given $\boldsymbol{\xi}$. If the set $\Pi(\boldsymbol{q})$ is nonempty and the second-stage problem (2.1.2) is feasible for at least one $\boldsymbol{x}$, then the function $Q(\cdot,\boldsymbol{\xi})$ is polyhedral.*

Proposition 9. *Assume that for a given $\boldsymbol{x}=\boldsymbol{x}_0$ and $\boldsymbol{\xi}\in\boldsymbol{\Xi}$ the value of $Q(\boldsymbol{x}_0,\boldsymbol{\xi})$ is finite.*

$$\mathcal{D}(\boldsymbol{x},\boldsymbol{\xi}) := \arg\max_{\boldsymbol{d}\in\Pi(\boldsymbol{q})} \boldsymbol{d}^T(\boldsymbol{h}-\boldsymbol{T}\boldsymbol{x})$$

is the set of optimal solutions of the dual problem (2.1.3.1). Then $Q(\cdot,\boldsymbol{\xi})$ is subdifferentiable at $\boldsymbol{x}_0$ and

$$\partial Q(\boldsymbol{x}_0,\boldsymbol{\xi}) = -\boldsymbol{T}^T\mathcal{D}(\boldsymbol{x}_0,\boldsymbol{\xi}).$$

The expected value function

$$\phi(\mathbf{x}) := \mathbf{E}[Q(\mathbf{x},\boldsymbol{\xi})]$$

will be considered.

Proposition 10. *Assume that the probability distribution of $\boldsymbol{\xi}$ has a finite support $\boldsymbol{\Xi}=\{\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_K\}$ and the expected recourse cost $\phi(\cdot)$ has a finite value in at least one point $\overline{\boldsymbol{x}}\in\mathbb{R}^n$. Then the recourse function $\phi(\cdot)$ is polyhedral, and for any $\boldsymbol{x}_0\in dom\,\phi$,*

$$\partial\phi(\boldsymbol{x}_0) = \sum_{k=1}^{K} p_k\partial Q(\boldsymbol{x}_0,\boldsymbol{\xi}_k).$$

We will assume that the expectation function $\phi(\cdot) := \mathbf{E}[Q(\cdot,\boldsymbol{\xi})]$ takes on a finite value in at least one point $\overline{\mathbf{x}}\in\mathbb{R}^n$. Using the Proposition 9 and 10 we have that for every $\mathbf{x}_0\in\operatorname{dom}\phi$,

$$\partial\phi(\mathbf{x}_0) = -\sum_{k=1}^{K} p_k\mathbf{T}_k^T\mathcal{D}(\mathbf{x}_0,\boldsymbol{\xi}_k),$$

where

$$\mathcal{D}(\mathbf{x}_0, \boldsymbol{\xi}_k) := \arg\max\{\mathbf{d}^T(\mathbf{h}_k - \mathbf{T}_k\mathbf{x}_0) : \mathbf{W}_k^T\mathbf{d} \le \mathbf{q}_k\}.$$

Theorem 11. *Let $\overline{\boldsymbol{x}}$ be a feasible solution of the two-stage problem (2.1.1)-(2.1.2), $\overline{\boldsymbol{x}} \in \mathcal{X}$ and $\phi(\overline{\boldsymbol{x}})$ is finite. Then $\overline{\boldsymbol{x}}$ is an optimal solution of the two-stage problem (2.1.1)-(2.1.2), if there exists $\boldsymbol{d}_k \in \mathcal{D}(\overline{\boldsymbol{x}}, \boldsymbol{\xi}_k), k = 1, \ldots, K$, and $\boldsymbol{u} \in \mathbb{R}^m$, such that*

$$\begin{aligned} \sum_{k=1}^{K} p_k \mathbf{T}_k^T \mathbf{d}_k + \mathbf{A}^T\mathbf{u} &\le \mathbf{c}, \\ \overline{\mathbf{x}}^T\left(\mathbf{c} - \sum_{k=1}^{K} p_k \mathbf{T}_k^T \mathbf{d}_k - \mathbf{A}^T\mathbf{u}\right) &= 0. \end{aligned} \tag{2.1.3.5}$$

For the large-scale linear programming formulation (2.1.1.3), by minimizing the Lagrangian,

$$\begin{aligned} &\mathbf{c}^T\mathbf{x} + \sum_{k=1}^{K} p_k \mathbf{q}_k^T\mathbf{y}_k - \mathbf{u}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) - \sum_{k=1}^{K} p_k \mathbf{d}_k^T(\mathbf{T}_k\mathbf{x} + \mathbf{W}_k\mathbf{y}_k - \mathbf{h}_k) \\ &= \left(\mathbf{c} - \mathbf{A}^T\mathbf{u} - \sum_{k=1}^{K} p_k \mathbf{T}_k^T\mathbf{d}_k\right)^T\mathbf{x} + \sum_{k=1}^{K} p_k(\mathbf{q}_k - \mathbf{W}_k^T\mathbf{d}_k)^T\mathbf{y}_k + \mathbf{b}^T\mathbf{u} + \sum_{k=1}^{K} p_k \mathbf{h}_k^T\mathbf{d}_k, \end{aligned}$$

with respect to $\mathbf{x} \ge 0$ and $\mathbf{y}_k \ge 0, k = 1, \ldots, K$, we have the dual problem of problem (2.1.1.3) in the following formulation:

$$\max_{\mathbf{u}, \mathbf{d}_1, \ldots, \mathbf{d}_k} \mathbf{b}^T\mathbf{u} + \sum_{k=1}^{K} p_k \mathbf{h}_k^T\mathbf{d}_k$$

$$\begin{aligned} \textit{subject to} \qquad & \mathbf{c} - \mathbf{A}^T\mathbf{u} - \sum_{k=1}^{K} p_k \mathbf{T}_k^T\mathbf{d}_k \ge 0, \\ & \mathbf{q}_k - \mathbf{W}_k^T\mathbf{d}_k \ge 0, \quad k = 1, \ldots, K. \end{aligned}$$

We can write the optimality conditions of Theorem 11 in the form:

$$\sum_{k=1}^{K} p_k \mathbf{T}_k^T\mathbf{d}_k + \mathbf{A}^T\mathbf{u} \le \mathbf{c},$$

$$\overline{\mathbf{x}}^T\left(\mathbf{c} - \sum_{k=1}^{K} p_k \mathbf{T}_k^T\mathbf{d}_k - \mathbf{A}^T\mathbf{u}\right) = 0,$$

$$\mathbf{q}_k - \mathbf{W}_k^T\mathbf{d}_k \ge 0, \quad k = 1, \ldots, K,$$

$$\overline{\mathbf{y}}_k^T(\mathbf{q}_k - \mathbf{W}_k^T\mathbf{d}_k) = 0, \quad k = 1, \ldots, K.$$

The last two conditions correspond to feasibility and optimality of multipliers of $\mathbf{d}_k$ as solutions of the dual problems.

2.2 Polyhedral Two-stage Problems

According to [45] we will write the two-stage stochastic programming problem in a more general way,

$$\min_{\mathbf{x}} \; f_1(\mathbf{x}) + \mathbf{E}[Q(\mathbf{x}, \omega)], \tag{2.2.1}$$

where $Q(\mathbf{x}, \omega)$ is the optimal value of the second-stage problem

$$\begin{aligned} &\min_{\mathbf{y}} \; f_2(\mathbf{y}, \omega) \\ \textit{subject to} \qquad &\mathbf{T}(\omega)\mathbf{x} + \mathbf{W}(\omega)\mathbf{y} = \mathbf{h}(\omega). \end{aligned} \tag{2.2.2}$$

We say that the above formulated two-stage problem is *polyhedral*, if the following holds:

- The function $f_1(\cdot)$ is *polyhedral*. There exists vectors $\mathbf{c}_j$ and scalars α_j, $j = 1, \ldots, J_1$, vectors $\mathbf{a}_k$ and scalars b_k, $k = 1, \ldots, K_1$, such that $f_1(\mathbf{x})$ can be written as:

$$f_1(\mathbf{x}) = \begin{cases} \max\limits_{1 \le j \le J_1} \; \alpha_j + \mathbf{c}_j^T \mathbf{x} & \text{if} \quad \mathbf{a}_k^T \mathbf{x} \le b_k, \qquad k = 1, \ldots, K_1, \\ +\infty & \text{otherwise,} \end{cases}$$

  and dom $f_1 = \{\mathbf{x} : \mathbf{a}_k^T \mathbf{x} \le b_k, k = 1, \ldots, K_1\}$ is nonempty. Because function $f_1(\cdot)$ is polyhedral, it is also convex and lower semicontinuous.

- Function f_2 is *random polyhedral*, which means that there exist random vectors $\mathbf{q}_j = q_j(\omega)$ and random scalars $\gamma_j = \gamma_j(\omega), j = 1, \ldots, J_2$, random vectors $\mathbf{v}_k = \mathbf{v}_k(\omega)$ and random scalars $r_k = r_k(\omega), k = 1, \ldots, K_2$, such that $f_2(\mathbf{y}, \omega)$ can be written as:

$$f_2(\mathbf{y}, \omega) = \begin{cases} \max\limits_{1 \le j \le J_2} \; \gamma_j(\omega) + \mathbf{q}_j^T(\omega)\mathbf{y} & \text{if} \quad \mathbf{v}_k^T(\omega)\mathbf{y} \le r_k(\omega), \; k = 1, \ldots, K_2, \\ +\infty & \text{otherwise,} \end{cases}$$

  and dom $f_2(\cdot, \omega)$ is nonempty for a.e. ω.

It is evident, that the linear two-stage model (2.1.1)-(2.1.2) is a special case of a polyhedral two-stage model.

2.2.1 Scenarios and Optimality Conditions

The source for this subsection is [45]. The expected value function $\phi(\mathbf{x}) := \mathbf{E}[Q(\mathbf{x}, \omega)]$ is considered. We assume, that the probability measure P has a finite support, so there is a finite number of elementary events ω_k with respective positive probabilities $p_k, k = 1, \ldots, K$. Then we have $\mathbf{E}[Q(\mathbf{x}, \omega)] = \sum_{k=1}^{K} p_k Q(\mathbf{x}, \omega_k)$. The expectation $\mathbf{E}[Q(\mathbf{x}, \omega)]$ equals the optimal value of the following problem for a given $\mathbf{x}$:

$$\min_{\mathbf{y}_1,\ldots,\mathbf{y}_K} \sum_{k=1}^{K} p_k f_2(\mathbf{y}_k, \omega_k)$$

$$\textit{subject to} \quad \mathbf{T}_k\mathbf{x} + \mathbf{W}_k\mathbf{y}_k = \mathbf{h}_k, \quad k = 1, \ldots, K, \tag{2.2.1.1}$$

where $(\mathbf{h}_k, \mathbf{T}_k, \mathbf{W}_k) = (\mathbf{h}(\omega_k), \mathbf{T}(\omega_k), \mathbf{W}(\omega_k))$. If for at least one $k \in \{1, \ldots, K\}$ the following set

$$\text{dom } f_2(\cdot, \omega_k) \cap \{\mathbf{y} : \mathbf{T}_k\mathbf{x} + \mathbf{W}_k\mathbf{y} = \mathbf{h}_k\}$$

is empty, then the second-stage problem is infeasible and hence problem (2.2.1.1) is infeasible and the optimal value is $+\infty$. The proofs of the following proposition and theorem could be found in [45].

Proposition 12. *Assume a finite support of the probability measure P and that the expectation function $\phi(\cdot) := \boldsymbol{E}[Q(\cdot, \omega)]$ has a finite value in at least one point $\boldsymbol{x} \in \mathbb{R}^n$. Then the function $\phi(\cdot)$ is polyhedral, and for any $\boldsymbol{x}_0 \in dom\,\phi$,*

$$\partial\phi(\mathbf{x}_0) = \sum_{k=1}^{K} p_k \partial Q(\mathbf{x}_0, \omega_k).$$

The Lagrangian of the second-stage problem (2.2.2) is

$$L(\mathbf{y}, \mathbf{d}; \mathbf{x}, \omega) := f_2(\mathbf{y}, \omega) + \mathbf{d}^T \left(\mathbf{h}(\omega) - \mathbf{T}(\omega)\mathbf{x} - \mathbf{W}(\omega)\mathbf{y}\right).$$

The infimum of the Lagrangian function is

$$\begin{aligned} \inf_{\mathbf{y}} L(\mathbf{y}, \mathbf{d}; \mathbf{x}, \omega) &= \mathbf{d}^T \left(\mathbf{h}(\omega) - \mathbf{T}(\omega)\mathbf{x}\right) + \inf_{\mathbf{y}} [f_2(\mathbf{y}, \omega) - \mathbf{d}^T\mathbf{W}(\omega)\mathbf{y}] \\ &= \mathbf{d}^T \left(\mathbf{h}(\omega) - \mathbf{T}(\omega)\mathbf{x}\right) - f_2^* \left(\mathbf{W}(\omega)^T\mathbf{d}, \omega\right), \end{aligned}$$

where $f_2^*(\cdot, \omega)$ is the conjugate of $f_2(\cdot, \omega)$. Therefore we can write the dual problem of (2.2.2) as

$$\max_{\mathbf{d}} [\mathbf{d}^T(\mathbf{h}(\omega) - \mathbf{T}(\omega)\mathbf{x}) - f_2^* \left(\mathbf{W}^T(\omega)\mathbf{d}, \omega\right)]. \tag{2.2.1.2}$$

If the optimal value $Q(\mathbf{x}, \omega)$ of the second-stage problem (2.2.2) is less than $+\infty$ for some $(\mathbf{x}, \omega)$, then it is equal to the optimal value of the dual problem (2.2.1.2). The set of optimal solutions of the dual problem (2.2.1.2) is denoted by $\mathcal{D}(\mathbf{x}, \omega)$. Optimality conditions for polyhedral two-stage problems and finite number of scenarios are given by the following theorem.

Theorem 13. *Assume that the probability measure P with a finite support. Then a point $\overline{\boldsymbol{x}}$ is an optimal solution of the first-stage problem (2.2.1), if there exists $\boldsymbol{d}_k \in \mathcal{D}(\overline{\boldsymbol{x}}, \omega_k), k = 1, \ldots, K$, such that*

$$0 \in \partial f_1(\overline{\mathbf{x}}) - \sum_{k=1}^{K} p_k \mathbf{T}_k^T \mathbf{d}_k.$$

2.3 General Formulation of the Two-stage Problem

As it has been already mentioned before, the value of the first-stage decision vector $\mathbf{x}$ has to be chosen *before* the realization of the unknown quantities, summarized in the data vector $\boldsymbol{\xi} = \boldsymbol{\xi}(\omega)$, is known. The value of the second part, $\mathbf{y}$, can be chosen *after* the realization of $\boldsymbol{\xi}$ becomes known and it is generally dependent on the realization of $\boldsymbol{\xi}$ and on the choice of the first-stage decision vector $\mathbf{x}$. At the first stage we have to solve the expectation optimization problem

$$\min_{\mathbf{x}\in\mathcal{Y}} \mathbf{E}[F(\mathbf{x},\omega)], \tag{2.3.1}$$

where $\mathcal{Y} \subset \mathbb{R}^n$. For the first stage problem we have

$$F(\mathbf{x},\omega) := \mathbf{c}^T\mathbf{x} + Q(\mathbf{x},\boldsymbol{\xi}(\omega)),$$

where $Q(\mathbf{x},\boldsymbol{\xi})$ is the optimal value of the second-stage optimization problem (2.1.2). Here, the explicit dependence on the second-stage decision variables $\mathbf{y}$ is suppressed. This subsection is adopted from [42] and also the proof of Proposition 14 could be found there.

Similarly to the problem (2.1.2.1), we can also relax the problem (2.3.1) - we allow the first-stage decision variables to depend on the random data and we correct for that by imputing anticipativity constraints. We will denote by $\mathcal{M} = \mathcal{M}(\Omega,\mathcal{F},\mathcal{Y})$ the space of measurable mappings $\mathbf{x}(\cdot) : \Omega \to \mathcal{Y}$, for which the expectation function $\mathbf{E}[F(\mathbf{x}(\omega),\omega)]$ is well defined. Here $\mathbf{x}(\cdot)$ is a vector valued function of ω. We can write the relaxed problem as:

$$\min_{\mathbf{x}(\cdot)\in\mathcal{M}} \mathbf{E}[F(\mathbf{x}(\omega),\omega)]. \tag{2.3.2}$$

The optimal value function of problem (2.3.1) is denoted by

$$\vartheta(\omega) := \inf_{\mathbf{x}\in\mathcal{Y}} F(\mathbf{x},\omega).$$

In problem (2.3.2), we minimize over all mappings $\mathbf{x}(\omega)$ in $\mathcal{M}$. If $\Omega := \{\omega_1,\ldots,\omega_K\}$ is finite with corresponding probabilities $p_1,\ldots,p_K$, then $\mathbf{x}(\omega)$ can be identified with $(\mathbf{x}_1,\ldots,\mathbf{x}_K)$, $\mathbf{x}_k := \mathbf{x}(\omega_k)$. Then we can write problem (2.3.2) in the form

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_K} \sum_{k=1}^{K} p_k F(\mathbf{x}_k,\omega_k). \tag{2.3.3}$$

Proposition 14. *Assume that the following holds:*

(i) the function $F(\boldsymbol{x},\omega)$ is random lower semicontinuous,

(ii) either $\boldsymbol{E}[\vartheta(\omega)_+] < +\infty$ or $\boldsymbol{E}[(-\vartheta(\omega))_+] < +\infty$.

Then

$$\inf_{\mathbf{x}(\cdot)\in\mathcal{M}} \mathbf{E}[F(\mathbf{x}(\omega),\omega)] = \mathbf{E}\left[\inf_{\mathbf{x}\in\mathcal{Y}} F(\mathbf{x},\omega)\right]. \tag{2.3.4}$$

Nonanticipativity constraints of problem (2.3.2) can be expressed as $\mathbf{x}(\omega) = \mathbf{E}[\mathbf{x}(\omega)]$ for all $\omega \in \Omega$. These constraints are an extension to constraints $\mathbf{x}_k = \sum_{i=1}^{K} p_i \mathbf{x}_i$, $k = 1, \ldots, K$. It is sufficient to verify the nonanticipativity constraints for P-almost every $\omega \in \Omega$, because the expected value of two random variables which may differ on a set of measure zero is the same. Using (2.3.2), Proposition 14 and the nonanticipativity constraints of problem (2.3.2) as they are stated above, we have that

$$\inf_{\mathbf{x}\in\mathcal{Y}} \mathbf{E}[F(\mathbf{x},\omega)] \geq \mathbf{E}\left[\inf_{\mathbf{x}\in\mathcal{Y}} F(\mathbf{x},\omega)\right]. \tag{2.3.5}$$

We can formulate the two-stage problem, where the second-stage decisions appear explicitly:

$$\min_{\mathbf{x}\in\mathcal{Y}} \mathbf{E}[V(\mathbf{x},\boldsymbol{\xi}(\omega)], \tag{2.3.6}$$

where $V(\mathbf{x},\boldsymbol{\xi}(\omega))$ is the optimal value of the second-stage problem

$$\min_{\mathbf{y}\in\mathcal{Z}} F(\mathbf{x},\mathbf{y},\boldsymbol{\xi}(\omega))$$

$$\textit{subject to} \qquad G_i(\mathbf{x},\mathbf{y},\boldsymbol{\xi}(\omega)) \leq 0, \qquad i = 1,\ldots,m. \tag{2.3.7}$$

Here we consider $\mathcal{Y} \subset \mathbb{R}^{n_1}$, $\mathcal{Z} \subset \mathbb{R}^{n_2}$, the objective function

$$F : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^d \to \mathbb{R}$$

and the constraint functionals

$$G_i : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^d \to \mathbb{R}, \qquad i = 1,\ldots,m.$$

We can formulate the above stated problem in an abstract form

$$\min_{\mathbf{x}\in\mathcal{Y},\mathbf{y}(\cdot)\in\mathcal{Z}} \mathbf{E}[F(\mathbf{x},\mathbf{y}(\omega),\boldsymbol{\xi}(\omega))] \tag{2.3.8}$$

$$\textit{subject to} \qquad G_i(\mathbf{x},\mathbf{y}(\omega),\boldsymbol{\xi}(\omega)) \leq 0, \qquad i = 1,\ldots,m. \tag{2.3.9}$$

$$\mathbf{x} \in \mathcal{Y}, \tag{2.3.10}$$

$$\mathbf{y}(\omega) \in \mathcal{Z}, \tag{2.3.11}$$

where $\mathcal{Y} := \mathbb{R}^{n_1}$ and $\mathcal{Z}$ is a space of measurable functions $\Omega \to \mathbb{R}^{n_2}$. The second-stage decision variables $\mathbf{y}(\omega)$ are viewed as a random vector in $\mathbb{R}^{n_2}$, whereas vector $\boldsymbol{\xi}(\omega)$ represents the random data of the problem with a given distribution. The inequalities (2.3.9) and the inclusion (2.3.11) have to hold for P a.e. $\omega \in \Omega$. The probability measure P on $(\Omega, \mathcal{F})$ generates the corresponding probability distribution of the random vector $(\boldsymbol{\xi}(\omega), \mathbf{y}(\omega))$, so "for $P-$a.e. $\omega \in \Omega$" in that context will mean that the event happens for a.e. realization of the random vector $(\boldsymbol{\xi}, \mathbf{y})$. In the formulation (2.3.8.) - (2.3.11) we allow the second-stage decisions $\mathbf{y}$ to be functions of the elementary event ω, therefore the functional space $\mathcal{Z}$ needs to be specified : the mappings $\mathbf{y} : \Omega \to \mathbb{R}^{n_2}$ have to be measurable with respect to the σ-algebra $\mathcal{F}$ such that the expectation (2.3.8) makes sense. Furthermore, since

$\mathbf{y}$ is a function of ω, the probability space $(\Omega, \mathcal{F}, P)$ can be identified with the probability space $(\mathbb{R}^d, \mathcal{B}, P_\xi)$ of the random vector $\boldsymbol{\xi}$ and $\mathbf{y}(\cdot)$ can be viewed as an element of a space of measurable mappings from $\mathbb{R}^d$ into $\mathbb{R}^{n_2}$. If $\boldsymbol{\xi}$ has a finite number of realizations, $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_K$, the sample space can be identified with the set $\Omega = \{1, \ldots, K\}$ equipped with the σ-algebra of all its subsets. It is sufficient to consider mappings $\mathbf{y} : \{1, \ldots, K\} \to \mathbb{R}^{n_2}$, which could be identified with vectors $\mathbf{y}_1, \ldots, \mathbf{y}_K \in \mathbb{R}^{n_2}$. The decision space in case of finitely many realizations is $\mathbb{R}^{n_1} \times \underbrace{\mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_2}}_{K\ times}$.

2.3.1 Value of Perfect Information

Following the paper [42], we will consider the two-stage stochastic programming problem as it was stated in (2.3.6) with $V(\mathbf{x}, \boldsymbol{\xi})$ the optimal value of the second-stage problem (2.3.7). In case the value of $\boldsymbol{\xi}$ is known at the time, when the first-stage decision should be made (we have a *perfect information* about the data $\boldsymbol{\xi}$), then the optimization problem becomes a deterministic problem

$$\min_{\mathbf{x} \in \mathcal{Y}} V(\mathbf{x}, \boldsymbol{\xi}). \tag{2.3.1.1}$$

We can write it in the equivalent form as

$$\begin{aligned} &\min_{\mathbf{x} \in \mathcal{Y}, \mathbf{y} \in \mathcal{Z}} F(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) \\ \textit{subject to} \quad &G_i(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) \leq 0, \qquad i = 1, \ldots, m. \end{aligned} \tag{2.3.1.2}$$

Both the optimal solution $\bar{\mathbf{x}}(\boldsymbol{\xi})$ (if it exists) and the optimal value $\upsilon(\boldsymbol{\xi})$ of problem (2.3.1.1) depend on the realization of the random data $\boldsymbol{\xi} = \boldsymbol{\xi}(\omega)$. The expected value

$$\mathbf{E}[\upsilon(\boldsymbol{\xi})] = \mathbf{E}\left[\inf_{\mathbf{x} \in \mathcal{Y}} V(\mathbf{x}, \boldsymbol{\xi}(\omega))\right], \tag{2.3.1.3}$$

is called the *wait-and-see* solution.

It is evident that for any $\mathbf{x} \in \mathcal{Y}$ and any $\boldsymbol{\xi}$ holds $V(\mathbf{x}, \boldsymbol{\xi}) \geq \upsilon(\boldsymbol{\xi})$. Therefore it holds that

$$\mathbf{E}[V(\mathbf{x}, \boldsymbol{\xi}(\omega))] \geq \mathbf{E}\left[\inf_{\mathbf{x} \in \mathcal{Y}} V(\mathbf{x}, \boldsymbol{\xi}(\omega))\right]. \tag{2.3.1.4}$$

Using (2.3.5) we have that

$$\inf_{\mathbf{x} \in \mathcal{Y}} \mathbf{E}[V(\mathbf{x}, \boldsymbol{\xi}(\omega))] \geq \mathbf{E}\left[\inf_{\mathbf{x} \in \mathcal{Y}} V(\mathbf{x}, \boldsymbol{\xi}(\omega))\right]. \tag{2.3.1.5}$$

The optimal value of the stochastic programming problem (2.3.6) is always greater than or equal to $\mathbf{E}[\upsilon(\boldsymbol{\xi})]$. We will suppose that problem (2.3.6) has an optimal solution $\hat{\mathbf{x}}$. We obtain that $V(\hat{\mathbf{x}}, \boldsymbol{\xi}) - \upsilon(\boldsymbol{\xi})$ is nonnegative for all $\boldsymbol{\xi}$, therefore its expected value is equal to zero if $V(\hat{\mathbf{x}}, \boldsymbol{\xi}) - \upsilon(\boldsymbol{\xi}) = 0$ with probability 1 (w.p. 1). For that reason, in (2.3.1.5) the equality holds if

$$V(\hat{\mathbf{x}}, \boldsymbol{\xi}(\omega)) = \inf_{\mathbf{x}\in\mathcal{Y}} V(\mathbf{x}, \boldsymbol{\xi}(\omega)) \qquad \text{for a.e. } \omega \in \Omega. \tag{2.3.1.6}$$

If there exists an optimal solution of (2.3.1.1) which does not depend on $\boldsymbol{\xi}$ w.p. 1., then the equality in (2.3.1.5) also holds.

Knowing the realization of $\boldsymbol{\xi}$, the *value of perfect information* is the difference $V(\hat{\mathbf{x}}, \boldsymbol{\xi}) - v(\boldsymbol{\xi})$. The *expected value of perfect information* is

$$\mathbf{EVPI} := \inf_{\mathbf{x}\in\mathcal{Y}} \mathbf{E}[V(\mathbf{x}, \boldsymbol{\xi}(\omega))] - \mathbf{E}\left[\inf_{\mathbf{x}\in\mathcal{Y}} V(\mathbf{x}, \boldsymbol{\xi}(\omega))\right]. \tag{2.3.1.7}$$

The inequality (2.3.1.5) gives us that **EVPI** is always nonnegative and it is equal to zero if condition (2.3.1.6) holds.

3. Multistage Stochastic Programs

3.1 General Description of Multistage Stochastic Programs

There are many situations in the real life, when decisions should be made at certain time points. Usually, the horizon and the sequence of times (stages) at which the decisions will be made are fixed. We require that the decisions made at any stage of the decision process are nonanticipative - they are allowed to depend only on the past observations and decisions. The sources for this chapter are [45], [10], [12], [14] and [42].

In the general *T-stage stochastic program* we consider a stochastic data process

$$\omega = (\omega_1, \ldots, \omega_{T-1})$$

and a decision process

$$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T).$$

The decisions $\mathbf{x}_2, \ldots, \mathbf{x}_T$ are assumed to be random vectors (not necessarily of the same dimension), whereas $\mathbf{x}_1$ is a nonrandom vector-valued variable. The random elements ω_i of ω may be of quite general nature, but they are mostly real random vectors. The realizations of ω are called *trajectories* or *scenarios*. As it was before, the probability distribution of ω is denoted by P and its support by Ω. The sequence of decisions and observations is

$$\mathbf{x}_1, \omega_1, \mathbf{x}_2, \omega_2, \ldots, \mathbf{x}_{T-1}, \omega_{T-1}, \mathbf{x}_T.$$

Furthermore, ω_T, which becomes known after the decision $\mathbf{x}_T$ is made, might contribute to the overall costs. The decision process is *nonanticipative*, which means that decisions taken at any stage of the process neither depend on future realizations of the stochastic data, nor on future decisions, whereas the past information and the probabilistic specification $(\Omega, \mathcal{F}, P)$ of the process ω are exploited. We emphasize, that even in case when ω_T contributes to the overall costs, none of the decisions $\mathbf{x}_t$ does depend on it. We assume that the probability distribution P is known and it does not depend on $\mathbf{x}$. We will denote by $\omega^{t-1,\bullet} := (\omega_1, \ldots, \omega_{t-1})$ the part of the stochastic data process that precedes the stage t and by $\mathbf{x}^{t-1,\bullet} := (\mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$ the sequence of decisions at stages $1, \ldots, t-1$. Therefore the decision at stage t is $\mathbf{x}_t = \mathbf{x}_t(\mathbf{x}^{t-1,\bullet}, \omega^{t-1,\bullet}, P)$. The marginal probability distribution of ω_t is denoted by P_t and its conditional probability distribution by $P_t(\cdot|\omega^{t-1,\bullet})$, $t = 2, \ldots, T-1$. We can express the dependence of the decisions only on the history and the probabilistic specification as follows: let the σ-field generated by the observations $\omega^{t-1,\bullet}$ be denoted by $\mathcal{F}_{t-1} \subseteq \mathcal{F}$. The dependence of the t^{th}-stage decision $\mathbf{x}_t$ only on the past observations means, that $\mathbf{x}_t$ is measurable with respect to $\mathcal{F}_{t-1}$, similarly, $\mathbf{x}_t$ is $\mathcal{F}_{t-1}$-adapted. In each stage the decisions are limited by constraints that might depend on the previous decisions and observations. Stages do not have to refer exactly to the time periods, they correspond to the steps in the decision process. The first-stage decisions consist of all decisions that have to be made before any further information is

known, while the second-stage decisions can adapt to the known information, etc. Let the importance of the first-stage decison be illustrated by the example of the decision about the capacity of a new water reservoir or the decision about an initial contract or allocation of funds. The main emphasis is on the first-stage decision, which includes all decisions that have to be selected before. In the next section, formulations of the multistage stochastic programs will be discussed.

3.2 Multistage Stochastic Programs Formulation

The outcome of the sequence $\mathbf{x}_1, \omega_1, \mathbf{x}_2, \omega_2, \ldots, \mathbf{x}_{T-1}, \omega_{T-1}, \mathbf{x}_T$ will be quantified by a function $f_0(\mathbf{x}, \omega)$. As it was before, our aim will be to minimize the expected value $\mathbf{E}[f_0(\mathbf{x}, \omega)]$ subject to some constraints. As it is in [12], we will consider given nonempty sets $\mathcal{X}_t$ in $\mathbb{R}^{n_t}, t = 1, \ldots, T$. The t^{th}-stage constraints are then denoted by

$$\mathcal{X}_t(\omega) = \{\mathbf{x}^{t,\bullet} \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_t : f_{ti}\left(\mathbf{x}^{t,\bullet}, \omega^{t-1,\bullet}\right) \leq 0, i = 1, \ldots, m_t\} \tag{3.2.1}$$

We can see from the form of the constraints (3.2.1), that the choice of decisions of $\mathcal{X}_t(\omega)$ is not constrained by future decisions or observations, but this does not generally exclude the presence of *induced constraints* - they must be fulfilled We suppose, that all functions are measurable with respect to ω and all expectations exist (this is especially fulfilled, if Ω is a finite set, and it is to guarantee the existence of feasible and nonanticipative decision process $\mathbf{x}$ for almost all ω). We also assume that relations including random elements hold with probability 1 and that all infima are attained, therefore we write *min* instead of *inf*.

3.2.1 Two Formulations of Multistage Stochastic Programming Problems

The aim of the *T-stage stochastic program* is to find

$$\begin{gathered} \mathbf{x}_1 \in \mathcal{X}_1 \text{ and } \mathbf{x}_t \ \mathcal{F}_{t-1}\text{-measurable}, \ \mathbf{x}^{t,\bullet} \in \mathcal{X}_t(\omega), t = 1, \ldots, T, \text{ a.s.} \\ \text{that minimizes } \mathbf{E}\left\{f_0\left(\mathbf{x}_1, \mathbf{x}_2(\mathbf{x}_1, \omega_1), \ldots, \mathbf{x}_T(\mathbf{x}^{T-1,\bullet}, \omega^{T-1,\bullet}), \omega\right)\right\}. \end{gathered} \tag{3.2.1.1}$$

Sometimes the realizations of ω_T (those behind the horizon) may contribute to the overall observed costs, but they do not affect the decision process (assumption of nonanticipativity). The decison process may be affected by the *probability distribution* of ω_T. If we choose the objective function f_0 in (3.2.1.1) as an indicator function of a certain interval, we obtain the *probability objective function* of the form

$$P\left[f_0\left(\mathbf{x}_1, \mathbf{x}_2(\mathbf{x}_1, \omega_1), \ldots, \mathbf{x}_T(\mathbf{x}^{T-1,\bullet}, \omega^{T-1,\bullet}), \omega\right) \notin \mathcal{I}\right]$$

where $\mathcal{I}$ is a given interval of desirable values of f_0. If we replace the requirement that the constraints $\mathbf{x}^{t,\bullet} \in \mathcal{X}_t(\omega)$ *a.s.*, $t = 1, \ldots, T$ by the requirement that $\mathbf{x}^{t,\bullet} \in$

$\mathcal{X}_t(\omega), t = 1, \ldots, T$, *holds true with a prescribed probability*, we have stochastic program with *probabilistic* or *chance constraints*.

The *second formulation of the T-stage stochastic program* is based on a recursive evaluation of the overall objective function. As it is in [14], this allows us to write the multistage stochastic program as a sequence of a nested two-stage programs. There were various schemes considered to reduce the T-stage stochastic program (3.2.1.1) to a sequence of similar t-stage programs, where $t < T$. If we do not consider ω_T, we can then define the objective functions recursively as

$$\begin{aligned} \varphi_T(\mathbf{x}^{T\bullet}, \omega^{T-1,\bullet}) &\equiv f_0(\mathbf{x}, \omega) \\ \varphi_t(\mathbf{x}^{t,\bullet}, \omega^{t-1,\bullet}) &= \mathbf{E}_{\omega_t|\omega^{t-1,\bullet}} \left[\min_{\mathbf{x}_{t+1}} \varphi_{t+1}(\mathbf{x}^{t+1,\bullet}, \omega^{t,\bullet}) \right], t = 2, \ldots, T-1 \\ \varphi_1(\mathbf{x}_1) &= \mathbf{E}_{\omega_1} \min_{\mathbf{x}_2} \varphi_2(\mathbf{x}^{2,\bullet}, \omega_1). \end{aligned} \tag{3.2.1.2}$$

The minimization is realized over the corresponding t^{th}-stage constraints. The symbol $\mathbf{E}_{\omega|\omega'}$ denotes the expectation with respect to ω conditioned by ω'.

For relating an optimal solution of the T-stage problem to those minimizing the t-stage objective functions φ_t, we need to meet some requirements - boundedness assumptions concerning sets defined by the t-stage constraints and convexity of f_0 as a function of $\mathbf{x}$. In that case not only the canonical projections of the optimal solution $\hat{\mathbf{x}}^{T,\bullet}$ of the T-stage problem are optimal solutions of the t-stage problems, $t < T$, but also the optimal solutions of the t-stage problems can be extended to an optimal solution of the T-stage problem. We will consider $\hat{\mathbf{x}}_1 \in \arg\min \varphi_1(\mathbf{x}_1)$ over the first-stage constraints $\mathbf{x}_1 \in \mathcal{X}_1$ and $f_{1i}(\mathbf{x}_1) \leq 0, i = 1, \ldots, m_1$. We will obtain the next component of the optimal solution, $\hat{\mathbf{x}}_2(\hat{\mathbf{x}}_1, \omega_1)$, by solving $\min_{\mathbf{x}_2} \varphi_2(\hat{\mathbf{x}}_1, \mathbf{x}_2, \omega_1)$ over the second-stage constraints, etc. To extend these results to problems which include ω_T, we introduce a fictious decision $\mathbf{x}_{T+1}$, which does not affect the value of the objective function f_0. As it is in [14], for

$$f_0(\mathbf{x}, \omega) = f_{10}(\mathbf{x}_1) + \sum_{t=2}^{T} f_{t0}(\mathbf{x}^{t-1,\bullet}, \mathbf{x}_t, \omega^{t-1,\bullet})$$

we can write the scheme (3.2.1.2) as a sequence of *nested two-stage stochastic programs* in the following form

$$\begin{aligned} &\min \mathbf{E}[f_0(\mathbf{x}, \omega)] := f_{10}(\mathbf{x}_1 + \mathbf{E}_{\omega_1}[\varphi_1(\mathbf{x}_1, \omega_1)] \\ &\textit{subject to} \qquad \mathbf{x}_1 \in \mathcal{X}_1 \text{ and } f_{1i}(\mathbf{x}_1) \leq 0, \qquad i = 1, \ldots, m_1, \end{aligned} \tag{3.2.1.3}$$

where for $t = 2, \ldots, T$, for given $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$ and observed realizations of $\omega_1, \ldots, \omega_{t-1}$, $\varphi_{t-1}(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \omega_1, \ldots, \omega_{t-1})$ is the optimal value of the stochastic program

$$\begin{aligned} &\min f_{t0}(\mathbf{x}_t, \omega^{t-1,\bullet}) + \mathbf{E}_{\omega_t|\omega^{t-1,\bullet}}[\varphi_t(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \omega_1, \ldots, \omega_{t-1}, \omega_t)] \\ &\textit{subject to} \qquad \mathbf{x}_t \in \mathcal{X}_t \qquad \text{and} \\ &\qquad f_{ti}(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \omega_1, \ldots, \omega_{t-1}) \leq 0, \qquad i = 1, \ldots, m_t. \end{aligned} \tag{3.2.1.4}$$

Here we have $\varphi_T \equiv 0$ or it is an explicitly given function of $\mathbf{x}_1, \ldots, \mathbf{x}_T, \omega_1, \ldots, \omega_T$, if we consider the contribution of ω_T to the overall costs. We assume that all constraints with random parameters hold almost surely.

Multistage stochastic linear program with recourse is a special case of (3.2.1.3) - (3.2.1.4), stated in [14]. All functions f with any indices are linear in the decision variables. It is formulated as follows:

$$\min \mathbf{c}_1^T \mathbf{x}_1 + \mathbf{E}_{\omega_1} \left[\varphi_1(\mathbf{x}_1, \omega_1) \right]$$

$$\begin{aligned} \textit{subject to} \qquad & \mathbf{A}_1 \mathbf{x}_1 = \mathbf{b}_1, \\ & \mathbf{l}_1 \leq \mathbf{x}_1 \leq \mathbf{u}_1, \end{aligned} \tag{3.2.1.5}$$

where the functions $\varphi_{t-1}, t = 2, \ldots, T$, are defined recursively as

$$\varphi_{t-1}(\mathbf{x}^{t-1,\bullet}, \omega^{t-1,\bullet}) = \min_{\mathbf{x}_t} \left\{ \mathbf{c}_t(\omega^{t-1,\bullet})^T \mathbf{x}_t + \mathbf{E}_{\omega_t | \omega^{t-1,\bullet}} \left[\varphi_t(\mathbf{x}^{t-1,\bullet}, \mathbf{x}_t, \omega^{t-1,\bullet}, \omega_t) \right] \right\}$$

$$\begin{aligned} \textit{subject to} \qquad & \mathbf{B}_t(\omega^{t-1,\bullet}) \mathbf{x}_{t-1} + \mathbf{A}_t(\omega^{t-1,\bullet}) \mathbf{x}_t = \mathbf{b}_t(\omega^{t-1,\bullet}) \\ & \mathbf{l}_t(\omega^{t-1,\bullet}) \leq \mathbf{x}_t \leq \mathbf{u}_t(\omega^{t-1,\bullet}) \end{aligned} \tag{3.2.1.6}$$

and $\varphi_T \equiv 0$ or a given function of $\mathbf{x}$ and ω. All equalities and inequalities hold almost surely. The random vector ω_{t-1} generates the coefficients $\mathbf{b}_t, \mathbf{c}_t$ and matrices $\mathbf{A}_t, \mathbf{B}_t$ in the decision problem of the t^{th}-stage, $t = 2, \ldots, T$. Matrices $\mathbf{A}_t$ are (m_t, n_t)-matrices and the remaining matrices and vectors have corresponding dimensions. We suppose that all expectations are well-defined. For the first stage, we know the values of all elements $\mathbf{b}_1, \mathbf{c}_1, \mathbf{A}_1$. The decision variable (vector) $\mathbf{x}_1$ corresponds to the first-stage. If the matrices $\mathbf{A}_t$ are known nonrandom for all t, we speak of a *fixed recourse*.

3.2.2 Linear Multistage Stochastic Programs

This subsection is based on [45] and [14]. As it was before, we will denote by $\mathbf{x}_1, \ldots, \mathbf{x}_T$ decision vectors corresponding to stages $1, \ldots, T$. We can express the linear programming problem as:

$$\begin{array}{lccccccccccc} \min & \mathbf{c}_1^T\mathbf{x}_1 & + & \mathbf{c}_2^T\mathbf{x}_2 & + & \mathbf{c}_3^T\mathbf{x}_3 & + & \ldots & + & \mathbf{c}_T^T\mathbf{x}_T & & \\ \textit{subject to} & \mathbf{A}_1\mathbf{x}_1 & & & & & & & & & = & \mathbf{b}_1, \\ & \mathbf{B}_2\mathbf{x}_1 & + & \mathbf{A}_2\mathbf{x}_2 & & & & & & & = & \mathbf{b}_2, \\ & & & \mathbf{B}_3\mathbf{x}_2 & + & \mathbf{A}_3\mathbf{x}_3 & & & & & = & \mathbf{b}_3, \\ & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ & & & & & & & \mathbf{B}_{T,}\mathbf{x}_{T-1} & + & \mathbf{A}_T\mathbf{x}_T & = & \mathbf{b}_T, \\ & \mathbf{x}_1 \geq 0, & & \mathbf{x}_2 \geq 0, & & \mathbf{x}_3 \geq 0, & & \ldots & & \mathbf{x}_T \geq 0. & & \end{array} \tag{3.2.2.1}$$

This problem can be viewed as a multistage stochastic programming problem, where for the first-stage, $\mathbf{c}_1, \mathbf{b}_1$ and $\mathbf{A}_1$ are known. For $t = 2, \ldots, T$, some or all of the vectors $\mathbf{c}_t, \mathbf{b}_t$ and matrices $\mathbf{A}_t, \mathbf{B}_t$ are random. We remind, that the random coefficients $\mathbf{A}_t, \mathbf{B}_t, \mathbf{b}_t, \mathbf{c}_t, t = 2, \ldots, T$ are generated by the random vector ω_{t-1}. We will denote by $\boldsymbol{\xi}_1 := (\mathbf{c}_1, \mathbf{A}_1, \mathbf{b}_1)$ the vector of (known) first-stage coefficients. Correspondingly, for $t = 2, \ldots, T$, the vector of random coefficients of the t^{th}-stage will be denoted by $\boldsymbol{\xi}_t := [\mathbf{c}_t(\omega_{t-1}), \mathbf{B}_t(\omega_{t-1}), \mathbf{A}_t(\omega_{t-1}), \mathbf{b}_t(\omega_{t-1})]$. Since it was

already mentioned several times, that the elements of the vector of coefficients $\boldsymbol{\xi}_t$ for $t = 2, \ldots, T$ are (some or all) random generated by the random vector ω_{t-1}, we will write shortly $\boldsymbol{\xi}_t := (\mathbf{c}_t, \mathbf{B}_t, \mathbf{A}_t, \mathbf{b}_t)$. In case of doubts, whether it is a random vector or its realization, it will be specified by putting ω_{t-1} as an argument of the coefficients.

Accordingly, we have the following sequence:

$$\mathbf{x}_1,\ \boldsymbol{\xi}_2,\ \mathbf{x}_2,\ \boldsymbol{\xi}_3,\quad \ldots\ldots\ldots\quad ,\ \boldsymbol{\xi}_T,\ \mathbf{x}_T.$$

Our aim remains the same - to design the decision process in such way, that the expected value of the total costs is minimal. The decision process is nonaticipative. We will assume $1 \leq t_1 \leq t_2 \leq T$, where t_1, t_2 denote some stages. Then the history of $\boldsymbol{\xi}$ from time t_1 to time t_2 is denoted by $\boldsymbol{\xi}_{[t_1,t_2]} := (\boldsymbol{\xi}_{t_1}, \ldots, \boldsymbol{\xi}_{t_2})$. Information available up to time t is correspondingly denoted by $\boldsymbol{\xi}_{[1,t]}$. We have from nonaticipativity that the decision vector $\mathbf{x}_t$ may depend on the information available up to time t (which is $\boldsymbol{\xi}_{[1,t]}$), but neither on further realizations of $\boldsymbol{\xi}$ nor on future decisions.

We will look at problem (3.2.2.1) from the view of the last stage T. At that stage, all realizations of $\boldsymbol{\xi}$ are known (we know $\boldsymbol{\xi}_{[1,T]}$) and the values of the earlier decision vectors, $\mathbf{x}_1, \ldots, \mathbf{x}_{T-1}$ have been already chosen. Therefore at the last stage T, we deal with the following simple linear programming problem:

$$\begin{aligned} \min_{\mathbf{x}_T} \quad & \mathbf{c}_T^T \mathbf{x}_T \\ \textit{subject to} \quad & \mathbf{B}_T \mathbf{x}_{T-1} + \mathbf{A}_T \mathbf{x}_T = \mathbf{b}_T, \\ & \mathbf{x}_T \geq 0. \end{aligned} \tag{3.2.2.2}$$

The optimal value of this problem is dependent on the decision vector at stage $T-1$, $\mathbf{x}_{T-1} \in \mathbb{R}^{n_{T-1}}$ and on the data $\boldsymbol{\xi}_T = (\mathbf{c}_T, \mathbf{B}_T, \mathbf{A}_T, \mathbf{b}_T)$ and it will be denoted by $\mathcal{Q}_T(\mathbf{x}_{T-1}, \boldsymbol{\xi}_T)$. We will have a look at stage $T-1$, when $\mathbf{x}_{T-2}$ and $\boldsymbol{\xi}_{[1,T-1]}$ are known. We have then the two-stage stochastic programming problem:

$$\begin{aligned} \min_{\mathbf{x}_{T-1}} \quad & \mathbf{c}_{T-1}^T \mathbf{x}_{T-1} + \mathbf{E}\left[\mathcal{Q}_T(\mathbf{x}_{T-1}, \boldsymbol{\xi}_T) \mid \boldsymbol{\xi}_{[1,T-1]}\right] \\ \textit{subject to} \quad & \mathbf{B}_{T-1} \mathbf{x}_{T-2} + \mathbf{A}_{T-1} \mathbf{x}_{T-1} = \mathbf{b}_{T-1}, \\ & \mathbf{x}_{T-1} \geq 0. \end{aligned} \tag{3.2.2.3}$$

The above stated problem depends on $\mathbf{x}_2 \in \mathbb{R}^{n_{T-2}}$ and on $\boldsymbol{\xi}_{[1,T-1]}$, its optimal value is denoted by $\mathcal{Q}_{T-1}(\mathbf{x}_{T-2}, \boldsymbol{\xi}_{[1,T-1]})$.

In general, at stage $t = 2, \ldots, T-1$, we are solving the problem:

$$\begin{aligned} \min_{\mathbf{x}_t} \quad & \mathbf{c}_t^T \mathbf{x}_t + \mathbf{E}\left[\mathcal{Q}_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t+1]}) \mid \boldsymbol{\xi}_{[1,t]}\right] \\ \textit{subject to} \quad & \mathbf{B}_t \mathbf{x}_{t-1} + \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t, \\ & \mathbf{x}_t \geq 0. \end{aligned} \tag{3.2.2.4}$$

The optimal value of the above stated problem is denoted by $\mathcal{Q}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]})$ and it is called *cost-to-go function*. The problem to find the first decisions, $\mathbf{x}_1$ is

$$\begin{aligned} \min_{\mathbf{x}_1} \quad & \mathbf{c}_1^T\mathbf{x}_1 + \mathbf{E}\left[\mathcal{Q}_2(\mathbf{x}_1, \boldsymbol{\xi}_2)\right] \\ \textit{subject to} \quad & \mathbf{A}_1\mathbf{x}_1 = \mathbf{b}_1, \\ & \mathbf{x}_1 \geq 0. \end{aligned} \tag{3.2.2.5}$$

All next stages t, $t = 2, \ldots, T$, are included in the above first-stage problem in the function $\mathcal{Q}_2(\mathbf{x}_1, \boldsymbol{\xi}_2)$ through the corresponding expected values. Since $\boldsymbol{\xi}_1$ is not random, function $\mathcal{Q}_2(\mathbf{x}_1, \boldsymbol{\xi}_2)$ does not depend on $\boldsymbol{\xi}_1$.

We say, that decisions $\bar{\mathbf{x}}_t(\boldsymbol{\xi}_{[1,t]})$ are *optimal*, if for $t = 1, \ldots, T$ the following condition holds for almost every realization of the random process

$$\bar{\mathbf{x}}_t(\boldsymbol{\xi}_{[1,t]}) \in \arg\min_{\mathbf{x}_t} \left\{\mathbf{c}_t^T\mathbf{x}_t + \mathcal{Q}_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t]}) : \mathbf{A}_t\mathbf{x}_t = \mathbf{b}_t - \mathbf{B}_t\bar{\mathbf{x}}_{t-1}(\boldsymbol{\xi}_{[1,t-1]}),\ \mathbf{x}_t \geq 0\right\}.$$

For $t = T$ we omit the term $\mathcal{Q}_{T+1}$ and for $t = 1$ we omit the term $\mathbf{B}_t\bar{\mathbf{x}}_{t-1}$. From what has been shown above (3.2.2.2) - (3.2.2.5), we arrived at the *nested formulation* of the linear multistage problem in the form

$$\min_{\mathcal{E}_1} \quad \mathbf{c}_1^T\mathbf{x}_1 + \mathbf{E}\left[\min_{\mathcal{E}_2} \quad \mathbf{c}_2^T\mathbf{x}_2 + \mathbf{E}\left[\ldots + \mathbf{E}\left[\min_{\mathcal{E}_T} \quad \mathbf{c}_T^T\mathbf{x}_T\right]\right]\right],$$

where $\mathcal{E}_1 = \{\mathbf{A}_1\mathbf{x}_1 = \mathbf{b}_1,\ \mathbf{x}_1 \geq 0\}$ and $\mathcal{E}_t = \{\mathbf{B}_t\mathbf{x}_{t-1} + \mathbf{A}_t\mathbf{x}_t = \mathbf{b}_t,\ \mathbf{x}_t \geq 0\}$ for $t = 2, \ldots, T$.

We say that the process $\{\boldsymbol{\xi}_t\}$ is *stagewise independent*, if $\boldsymbol{\xi}_t$ is stochastically independent of $\boldsymbol{\xi}_{[1,t-1]}$, $t = 2, \ldots, T$. Random process $\{\boldsymbol{\xi}_t\}$ is said to be *Markovian*, if for each $t = 2, \ldots, T-1$ the conditional distribution of $\boldsymbol{\xi}_t$ given $\boldsymbol{\xi}_{[t-1]}$ is the same as the conditional distribution of $\boldsymbol{\xi}_t$ given $\boldsymbol{\xi}_{t-1}$. Obviously, if the process is stagewise independent, then it is Markovian. If the process $\{\boldsymbol{\xi}_t\}$ is Markovian, our model simplifies - for given $\boldsymbol{\xi}_{T-1}$, the conditional expectation in problem (3.2.2.3) is independent of $\boldsymbol{\xi}_{[1,T-2]}$ and therefore the optimal value of problem (3.2.2.3) depends only on $\mathbf{x}_{T-2}$ and $\boldsymbol{\xi}_{T-1}$. Analogous to that, at stages $t = 2, \ldots, T-1$ the optimal value of problem (3.2.2.4) depends on $\mathbf{x}_{t-1}$ and $\boldsymbol{\xi}_t$ and we denote it by $\mathcal{Q}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t)$. Then we call $\boldsymbol{\xi}_t$ the *information state* of the model. Furthermore, if the process $\{\boldsymbol{\xi}_t\}$ is stagewise independent, then the expectation function $\mathcal{Q}_t$ does not depend on realizations of the random process, and we can write only $\mathcal{Q}_t(\mathbf{x}_{t-1})$, $t = 2, \ldots, T$.

We will now suppose a model with a full lower block triangular constraint matrix:

$$\begin{aligned} \min \quad & \mathbf{c}_1^T\mathbf{x}_1 + \mathbf{c}_2^T\mathbf{x}_2 + \mathbf{c}_3^T\mathbf{x}_3 + \ldots + \mathbf{c}_T^T\mathbf{x}_T \\ \textit{subject to} \quad & \mathbf{A}_{11}\mathbf{x}_1 = \mathbf{b}_1, \\ & \mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 = \mathbf{b}_2, \\ & \mathbf{A}_{31}\mathbf{x}_1 + \mathbf{A}_{32}\mathbf{x}_2 + \mathbf{A}_{33}\mathbf{x}_3 = \mathbf{b}_3, \\ & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ & \mathbf{A}_{T1}\mathbf{x}_1 + \mathbf{A}_{T2}\mathbf{x}_2 + \ldots + \mathbf{A}_{T,T-1}\mathbf{x}_{T-1} + \mathbf{A}_{TT}\mathbf{x}_T = \mathbf{b}_T, \\ & \mathbf{x}_1 \geq 0, \quad \mathbf{x}_2 \geq 0, \quad \mathbf{x}_3 \geq 0, \quad \ldots \quad \mathbf{x}_T \geq 0. \end{aligned} \tag{3.2.2.6}$$

In that case each subproblem depends on the entire history of our decisions, $\mathbf{x}_{[1,t-1]} := (\mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$. It has the form:

$$\begin{array}{lr}\min\limits_{\mathbf{x}_t} & \mathbf{c}_t^T\mathbf{x}_t + \mathbf{E}\left[\mathcal{Q}_{t+1}(\mathbf{x}_{[1,t]}, \boldsymbol{\xi}_{[1,t+1]}) \mid \boldsymbol{\xi}_{[1,t]}\right] \\ \textit{subject to} & \mathbf{A}_{t1}\mathbf{x}_1 + \ldots + \mathbf{A}_{t,t-1}\mathbf{x}_{t-1} + \mathbf{A}_{t,t}\mathbf{x}_t = \mathbf{b}_t, \\ & \mathbf{x}_t \geq 0.\end{array} \tag{3.2.2.7}$$

The optimal value of the subproblem (3.2.2.7) is denoted by $\mathcal{Q}_t(\mathbf{x}_{[1,t-1]}, \boldsymbol{\xi}_{[1,t]})$.

Sometimes it is good to transform the lower triangular formulation into the staircase formulation, which was shown at the beginning of this subsection. This could be done by including additional variables $\mathbf{r}_t$, that summarize the relevant history of our decisions. These additional variables are called *model state variables*. We call *model state equations* the relations which describe the next values of the state variables as a function of the current values of these variables, current decisions and current random parameters. In the problem (3.2.2.6) the vectors $\mathbf{x}_{[1,t]} = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ are sufficient model state variables. At each stage, they are updated to the linear state equation $\mathbf{x}_{[1,t]} = (\mathbf{x}_{[1,t-1]}, \mathbf{x}_t)$. We can write the constraint in (3.2.2.7) formally as

$$[\mathbf{A}_{t1}\mathbf{A}_{t2}\ldots\mathbf{A}_{t,t-1}]\mathbf{x}_{[1,t-1]} + \mathbf{A}_{t,t,}\mathbf{x}_t = \mathbf{b}_t.$$

For many problems it is possible to define model state variables of a reasonable size. Let the following structure be considered

$$\begin{array}{lcccccccccc}\min & \mathbf{c}_1^T\mathbf{x}_1 & + & \mathbf{c}_2^T\mathbf{x}_2 & + & \mathbf{c}_3^T\mathbf{x}_3 & +\ldots & + & \mathbf{c}_T^T\mathbf{x}_T & & \\ \textit{subject to} & \mathbf{A}_{11}\mathbf{x}_1 & & & & & & & & = & \mathbf{b}_1, \\ & \mathbf{B}_1\mathbf{x}_1 & + & \mathbf{A}_{22}\mathbf{x}_2 & & & & & & = & \mathbf{b}_2, \\ & \mathbf{B}_1\mathbf{x}_1 & + & \mathbf{B}_2\mathbf{x}_2 & + & \mathbf{A}_{33}\mathbf{x}_3 & & & & = & \mathbf{b}_3, \\ & \cdots & & & & & & & & & \\ & \mathbf{B}_1\mathbf{x}_1 & + & \mathbf{B}_2\mathbf{x}_2 & + & \ldots & +\mathbf{B}_{T-1}\mathbf{x}_{T-1} & + & \mathbf{A}_{TT}\mathbf{x}_T & = & \mathbf{b}_T, \\ & & & \mathbf{x}_1 \geq 0, & & \mathbf{x}_2 \geq 0, & \mathbf{x}_3 \geq 0, & \ldots & \mathbf{x}_T \geq 0, & & \end{array} \tag{3.2.2.8}$$

in which all blocks $\mathbf{A}_{it}, i = 2, \ldots, T$ are identical and observed at time t. For that case the state variables $\mathbf{r}_t, t = 2, \ldots, T$ can be defined recursively by the state equation $\mathbf{r}_t = \mathbf{r}_{t+1} + \mathbf{B}_t\mathbf{x}_t, t = 1, \ldots, T-1$, where $\mathbf{r}_0 = 0$. Then the problem (3.2.2.7) is noticeably simpler:

$$\begin{array}{ll}\min\limits_{\mathbf{x}_t,\mathbf{r}_t} & \mathbf{c}_t^T\mathbf{x}_t + \mathbf{E}\left[\mathcal{Q}_{t+1}(\mathbf{r}_t, \boldsymbol{\xi}_{[1,t+1]}) \mid \boldsymbol{\xi}_{[1,t]}\right] \\ \textit{subject to} & \mathbf{r}_{t-1} + \mathbf{A}_{tt}\mathbf{x}_t = \mathbf{b}_t, \\ & \mathbf{r}_t = \mathbf{r}_{t-1} + \mathbf{B}_t\mathbf{x}_t, \\ & \mathbf{x}_t \geq 0.\end{array}$$

The optimal value of the above stated subproblem is denoted by $\mathcal{Q}_t(\mathbf{r}_{t-1}, \boldsymbol{\xi}_{[1,t]})$ and it depends on $\mathbf{r}_{t-1}$.

Simple sign constraints $\mathbf{x}_t \geq 0$ can be replaced by a general constraint $\mathbf{x}_t \in \mathbf{X}_t$, where $\mathbf{X}_t$ is a convex polyhedron, which is defined by some linear equations and inequalities (they are local for stage t). The set $\mathbf{X}_t$ might be random, but it has to be known at stage t.

3.2.3 Convex Multistage Problems

According to [45], we can write the T-stage stochastic programming problem in the nested formulation very generally as

$$\min_{\mathbf{x}_1 \in \mathcal{X}_1} f_1(\mathbf{x}_1) + \mathbf{E}\left[\inf_{\mathbf{x}_2 \in \mathcal{X}_2(\mathbf{x}_1, \boldsymbol{\xi}_2)} f_2((\mathbf{x}_2, \boldsymbol{\xi}_2) + \mathbf{E}\left[\ldots + \mathbf{E}\left[\inf_{\mathbf{x}_T \in \mathcal{X}_T(\mathbf{x}_{T-1}, \boldsymbol{\xi}_T)} f_T(\mathbf{x}_T, \boldsymbol{\xi}_T)\right]\right]\right], \tag{3.2.3.1}$$

We remind, that $\mathbf{x}_t \in \mathbb{R}^{n_t}$ are decision variables, $\boldsymbol{\xi}_t$ is a vector if random coefficients , $f_t : \mathbb{R}^{n_t} \times \mathbb{R}^{d_t} \to \mathbb{R}$ are continuous functions and $\mathcal{X}_t : \mathbb{R}^{n_{t-1}} \times \mathbb{R}^{d_t} \to \mathbb{R}^{n_t}$, $t = 2, \ldots, T$ are measurable closed valued multifunctions. The first stage data, $\boldsymbol{\xi}_1, f_1 : \mathbb{R}^{n_1} \to \mathbb{R}$ and the set $\mathcal{X} \subset \mathbb{R}^{n_1}$ are deterministic. For the linear case, we had especially:

$$\begin{gathered} f_t(\mathbf{x}_t, \boldsymbol{\xi}_t) := \mathbf{c}_t^T \mathbf{x}_t, \quad \mathcal{X}_1 := \{\mathbf{x}_1 : \mathbf{A}_1 \mathbf{x}_1 = \mathbf{b}_1, \quad \mathbf{x}_1 \geq 0\} \\ \mathcal{X}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t) := \{\mathbf{x}_t : \mathbf{B}_t \mathbf{x}_{t-1} + \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t, \quad \mathbf{x}_t \geq 0\}, \quad t = 2, \ldots, T. \end{gathered}$$

As it was in the previous subsection, $\boldsymbol{\xi}$ the vector of the coefficients, e.g. $\boldsymbol{\xi}_t(\omega) := (\mathbf{c}_t(\omega_{t-1}), \mathbf{A}_t(\omega_{t-1}), \mathbf{B}_t(\omega_{t-1}), \mathbf{b}_t(\omega_{t-1})), t = 2, \ldots, T$. For the first-stage $\xi_1 := (\mathbf{c}_1, \mathbf{A}_1, \mathbf{b}_1)$ is deterministic.

We will now consider multistage problems of the form (3.2.3.1), with

$$\mathcal{X}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t) := \{\mathbf{x}_t : \mathbf{B}_t \mathbf{x}_t + \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t\}, \ t = 2, \ldots, T,$$

$\mathcal{X}_1 := \{\mathbf{x}_1 : \mathbf{A}_1 \mathbf{x}_1 = \mathbf{b}_1\}$ and $f_t(\mathbf{x}_t, \boldsymbol{\xi}_t), t = 1, \ldots, T$, are random lower semicontinuous functions. It is supposed that functions $f_t(\cdot, \boldsymbol{\xi}_t)$ are *convex* for almost every $\boldsymbol{\xi}_t$. For

$$f_t(\mathbf{x}_t, \boldsymbol{\xi}_t) := \begin{cases} \mathbf{c}_t^T \mathbf{x}_t & \text{if} \quad \mathbf{x}_t \geq 0, \\ +\infty & \text{otherwise}, \end{cases}$$

we have the linear multistage problem given in the nested formulation.

Here we solve at each stage the problem

$$\mathcal{Q}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]}) = \inf_{\mathbf{x}_t} \left\{ f_t(\mathbf{x}_t, \boldsymbol{\xi}_t) + \mathcal{Q}_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t]}) : \mathbf{B}_t \mathbf{x}_{t-1} + \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t \right\}, \tag{3.2.3.2}$$

where

$$\mathcal{Q}_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t]}) := \mathbf{E}\left[\mathcal{Q}_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t+1]}) \mid \boldsymbol{\xi}_{[t]}\right].$$

The function $Q_t(\cdot, \boldsymbol{\xi}_{[1,t]})$ is convex for every $t = 1, \ldots, T$. In fact,

$$\mathcal{Q}_T(\mathbf{x}_{T-1}, \boldsymbol{\xi}_T) = \inf_{\mathbf{x}_T} \phi(\mathbf{x}_T, \mathbf{x}_{T-1}, \boldsymbol{\xi}_T),$$

where

$$\phi(\mathbf{x}_T, \mathbf{x}_{T-1}, \boldsymbol{\xi}_T) := \begin{cases} f_T(\mathbf{x}_T, \boldsymbol{\xi}_T) & \text{if} \quad \mathbf{B}_T \mathbf{x}_{T-1} + \mathbf{A}_T \mathbf{x}_T = \mathbf{b}_T, \\ +\infty & \text{otherwise}, \end{cases}$$

We have that function $f_T(\cdot, \boldsymbol{\xi}_T)$ is convex because of that we have of function $\phi(\cdot, \cdot, \boldsymbol{\xi}_T)$, therefore the optimal value function $\mathcal{Q}_T(\cdot, \boldsymbol{\xi}_T)$ is also convex. By induction we can show the convexity of functions $\mathcal{Q}_t(\cdot, \boldsymbol{\xi}_{[1,t]})$ for $t = T, \ldots, 1$.

3.3 Optimality conditions

Inspired by [45], we will consider the cost-to-go functions $\mathcal{Q}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]})$ as they are defined in (3.2.3.2). The Lagrangian associated with the optimization problem in (3.2.3.2) is:

$$L_t(\mathbf{x}_t, \mathbf{d}_t, \mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]}) := f_t(\mathbf{x}_t, \boldsymbol{\xi}_t) + \mathcal{Q}_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t]}) + \mathbf{d}_t^T(\mathbf{b}_t - \mathbf{B}_t\mathbf{x}_{t-1} - \mathbf{A}_t\mathbf{x}_t).$$

We will define $\psi_t(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t]}) := f_t(\mathbf{x}_t, \boldsymbol{\xi}_t) + \mathcal{Q}_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t]})$. Then we have

$$\begin{aligned}\inf_{\mathbf{x}_t} L_t(\mathbf{x}_t, \mathbf{d}_t, \mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]}) &= -\sup_{\mathbf{x}_t}\left\{\mathbf{d}_t^T\mathbf{A}_t\mathbf{x}_t - \psi_t(\mathbf{x}_t, \boldsymbol{\xi}_{[1,t]})\right\} + \mathbf{d}_t^T(\mathbf{b}_t - \mathbf{B}_t\mathbf{x}_{t-1}) \\ &= -\psi_t^*(\mathbf{A}_t^T\mathbf{d}_t, \boldsymbol{\xi}_{[1,t]}) + \mathbf{d}_t^T(\mathbf{b}_t - \mathbf{B}_t\mathbf{x}_{t-1}),\end{aligned}$$

where $\psi_t^*(\cdot, \boldsymbol{\xi}_{[1,t]})$ is the conjugate function of $\psi_t(\cdot, \boldsymbol{\xi}_{[1,t]})$. The Lagrangian dual of the optimization problem on the right-hand side of (3.2.3.2) can be written as:

$$\max_{\mathbf{d}_t}\left\{-\psi_t^*(\mathbf{A}_t^T\mathbf{d}_T, \boldsymbol{\xi}_{[1,t]}) + \mathbf{d}_t^T(\mathbf{b}_t - \mathbf{B}_t\mathbf{x}_{t-1})\right\}. \tag{3.3.1}$$

Optimization problems (3.2.3.2) and (3.3.1) are convex. Under certain regularity conditions, there is no duality gap between these two problems (the optimal values of the two problems are equal to each other). Particularly, we can formulate these two conditions:

(C1) The functions $f_t(\mathbf{x}_t, \boldsymbol{\xi}_t), t = 1, \ldots, T$, are random polyhedral, and we have a finite number of scenarios.

(C2) There is a neighborhood of $\mathbf{b}_t$, such that for any $\mathbf{b}_t^{'}$ in that neighborhood the optimal value of the problem (3.2.3.2) with $\mathbf{b}_t$ replaced by $\mathbf{b}_t^{'}$ is finite.

The set of optimal solutions of (3.3.1) will be denoted as $\mathcal{D}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]})$. All subdifferentials in the next statements will be taken with respect to $\mathbf{x}_t$ for a corresponding $t = 1, \ldots, T$. The proof of the next proposition can be found in [45].

Proposition 15. *Assume that either condition* (C1) *holds and* $\mathcal{Q}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]})$ *is finite or condition* (C2) *holds. Then the following holds:*

(i) *there is no duality gap between problems* (3.2.3.2) *and* (3.3.1),

$$\mathcal{Q}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]}) = \sup_{\mathbf{d}_t}\left\{-\psi_t^*(\mathbf{A}_t^T\mathbf{x}_t, \boldsymbol{\xi}_{[1,t]}) + \mathbf{d}_t^T(\mathbf{b}_t - \mathbf{B}_t\mathbf{x}_{t-1})\right\},$$

(ii) $\bar{\mathbf{x}}_t$ *is an optimal solution of* (3.2.3.2) *if there exists* $\bar{\mathbf{d}}_t = \bar{\mathbf{d}}_t(\boldsymbol{\xi}_{[1,t]})$ *such that* $\bar{\mathbf{d}}_t \in \mathcal{D}(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]})$ *and*

$$0 \in \partial L_t(\bar{\mathbf{x}}_t, \bar{\mathbf{d}}_t),$$

(iii) *the function* $\mathcal{Q}_t(\cdot, \boldsymbol{\xi}_{[1,t]})$ *is subdifferentiable at* $\mathbf{x}_{t-1}$ *and*

$$\partial\mathcal{Q}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[t-1]}) = -\mathbf{B}_t^T\mathcal{D}_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_{[1,t]}).$$

4. Scenario Approach to Multistage Stochastic Programs

Similarly to as it was in the case of two-stage stochastic programs, also in the case of multistage stochastic programs we often approximate the true probability P of the stochastic process ω by a discrete probability distribution carried by a finite number of atoms, denoted by $\omega^1, \ldots, \omega^K$. The supports of marginal probability distributions P_t are finite sets and also the supports of the conditional probability distributions $P_t(\cdot, \mid \omega^{t-1,\bullet})$ are finite sets for $\forall\, t$. We will denote by $\mathcal{S}_t(\omega^{t-1,\bullet})$ the supports of the conditional probability distributions of ω_t conditioned by past realizations $\omega^{t-1,\bullet} = (\omega_1, \ldots, \omega_{t-1})$. We call *scenarios at stage* t the sequences of realizations $\omega^{t,\bullet} = (\omega_1, \ldots, \omega_t)$. Let $\mathcal{K}_t, t = 2, \ldots, T$ be disjoint sets of indices, all possible realizations of $\omega^{t-1,\bullet}$ are listed as $\tilde{\omega}_{k_t}, k_t \in \mathcal{K}_t$. The sources for this chapter are [45], [10], [12], [14] and [42].

4.1 Scenario Based Formulations of Multistage Stochastic Programs

Let the corresponding values of the t^{th}-stage coefficients be denoted by the same subscripts of all possible realizations of $\omega^{t-1,\bullet}$. The total number of scenarios K is equal to the total number of elements of $\mathcal{K}_T$. Each scenario $\omega^k = (\omega_1^k, \ldots, \omega_{T-1}^k)$ generates a sequence of coefficients $\{\mathbf{c}_{k_2}, \ldots, \mathbf{c}_{k_T}\}, \{\mathbf{A}_{k_2}, \ldots, \mathbf{A}_{k_T}\}, \{\mathbf{B}_{k_2}, \ldots, \mathbf{B}_{k_T}\}$, $\{\mathbf{b}_{k_2}, \ldots, \mathbf{b}_{k_T}\}, \{\mathbf{l}_{k_2}, \ldots, \mathbf{l}_{k_T}\}$ and $\{\mathbf{u}_{k_2}, \ldots, \mathbf{u}_{k_T}\}$. The vector of feasible solutions of the scenario ω^k subproblem is denoted by $\mathbf{x}(\omega^k) := (\mathbf{x}_1, \mathbf{x}_{k_2}, \ldots, \mathbf{x}_{k_T})$. The set of constraints for scenario ω^k according to [14] is

$$
\begin{aligned}
\mathbf{A}_1\mathbf{x}_1 &= \mathbf{b}_1,\\
\mathbf{B}_{k_2}\mathbf{x}_1 + \mathbf{A}_{k_2}\mathbf{x}_{k_2} &= \mathbf{b}_{k_2},\\
\mathbf{B}_{k_3}\mathbf{x}_{k_2} + \mathbf{A}_{k_3}\mathbf{x}_{k_3} &= \mathbf{b}_{k_3},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
\mathbf{B}_{k_T}\mathbf{x}_{T-1} + \mathbf{A}_{k_T}\mathbf{x}_{T_2} &= \mathbf{b}_{k_T},\\
\\
\mathbf{x}_1 \le \mathbf{u}_1,\ \mathbf{l}_{k_2} \le \mathbf{x}_{k_2} \le \mathbf{u}_{k_2},\ \ldots\ ,\ \mathbf{l}_{k_T} &\le \mathbf{x}_{k_T} \le \mathbf{u}_{k_T}.
\end{aligned}
\tag{4.1.1}
$$

We can write the multistage stochastic programs with linear constraints as a large-scale deterministic program. For a given scenario ω^k, the vector composed of all corresponding objective function coefficients to that scenario, $\mathbf{c}_1, \mathbf{c}_{k_t}, t = 2, \ldots, T$ is denoted by $\mathbf{c}(\omega^k)$. We will denote the matrix of all coefficients of the system of constraints in (4.1.1) as $\mathbf{A}(\omega^k)$ for scenario ω^k, the vector of the right-hand sides as $\mathbf{b}(\omega^k)$ and as $\mathbf{l}(\omega^k)$ and $\mathbf{u}(\omega^k)$ the vectors of the lower and upper bounds. The corresponding decision vector $\mathbf{x}(\omega^k)$ is composed of stage related subvectors $\mathbf{x}_t(\omega^k)$ for $\forall t$. The nonanticipativity constraints will be of the form $\mathbf{x}_1(\omega^k) = \mathbf{x}_1(\omega^{k'}), \forall k, k'$, for the first-stage decisions and similarly to that for the t^{th}-stage we also need to guarantee that the t^{th}-stage decisions based on the same

history are equal. We can express the constraints in the form $\mathbf{x} = \mathbf{U}\mathbf{x}$, where $\mathbf{x}$ consists of carefully grouped components of all decision vectors $\mathbf{x}(\omega^k)$ and $\mathbf{U}$ is a 0-1 matrix of coefficients of the nonanticipativity constraints. We obtain the *scenario-splitted form* of the T-stage stochastic linear program as it is given in [10]:

$$\min_{\mathcal{X}_D \cap \mathcal{C}} \left\{ \sum_{k=1}^{K} p^k \mathbf{c}(\omega^k)^T \mathbf{x}(\omega^k) \mid \mathbf{A}(\omega^k)\mathbf{x}(\omega^k) = \mathbf{b}(\omega^k), \mathbf{l}(\omega^k) \leq \mathbf{x}(\omega^k) \leq \mathbf{u}(\omega^k), \forall k \right\}. \tag{4.1.2}$$

The set $\mathcal{X}_D$ is the set defined by deterministic constraints on $\mathbf{x}_t(\omega^k)$ for $\forall t, k$ and the set $\mathcal{C}$ is the set defined by the nonanticipativity conditions. The probabilities of each scenario ω^k are denoted by p^k.

Another scenario-based formulation of multistage stochastic linear program is characterized by implicit inclusion of nonanticipativity constraints and data organized in the form of a *scenario tree*. The nodes are determined by all considered realizations of $\tilde{\omega}_{k_t}, k_t \in \mathcal{K}_t, t = 2, \ldots, T$ and by the root k_1. Each value $\tilde{\omega}_{k_{t+1}}$ of $\omega^{t,\bullet}, t = 1, \ldots, T$ has an immediate ancestor $\tilde{\omega}_{k_t}$ (the value of the corresponding $\omega^{t-1,\bullet}$). This unique ancestor is denoted by $a(k_{t+1})$, e.g. $a(k_2) = 1$ for all realizations of $\tilde{\omega}_{k_2}$ of the component ω_1. According to [14], we write the *T-stage stochastic linear program with recourse and with a finite number of scenarios in the arborescent form* as:

$$\min \quad \mathbf{c}_1^T \mathbf{x}_1 + \sum_{k_2=2}^{K_2} p_{k_2} \mathbf{c}_{k_2}^T \mathbf{x}_{k_2} + \sum_{k_3=K_2+1}^{K_3} p_{k_3} \mathbf{c}_{k_3}^T \mathbf{x}_{k_3} + \ldots + \sum_{k_T=K_{T-1}+1}^{K_T} p_{k_T} \mathbf{c}_{k_T}^T \mathbf{x}_{k_T}$$

subject to

$$\begin{array}{rrrrrrrrrrl}
\mathbf{A}_1\mathbf{x}_1 & & & & & & & & = \mathbf{b}_1, & \\
\mathbf{B}_{k_2}\mathbf{x}_1 & + & \mathbf{A}_{k_2}\mathbf{x}_{k_2} & & & & & & = \mathbf{b}_{k_2}, & k_2 \in \mathcal{K}_2, \\
 & & \mathbf{B}_{k_3}\mathbf{x}_{a_{k_3}} & + & \mathbf{A}_{k_3}\mathbf{x}_{k_3} & & & & = \mathbf{b}_{k_3}, & k_3 \in \mathcal{K}_3, \\
 & & \cdot & & \cdot & & & & \cdot & \\
 & & & \cdot & & \cdot & & & \cdot & \\
 & & & & \cdot & & \cdot & & \cdot & \\
 & & & & & & \mathbf{B}_{k_T}\mathbf{x}_{a_{k_T}} & + \mathbf{A}_{k_T}\mathbf{x}_{k_T} & = \mathbf{b}_{k_T}, & k_T \in \mathcal{K}_T,
\end{array}$$

$$\mathbf{l}_1 \leq \mathbf{x}_1 \leq \mathbf{u}_1, \; \mathbf{l}_{k_t} \leq \mathbf{x}_{k_t} \leq \mathbf{u}_{k_t}, \; k_t \in \mathcal{K}_t, t = 2, \ldots, T, \tag{4.1.3}$$

with $K_1 = 1, \mathcal{K}_t = \{K_{t-1} + 1, \ldots, K_t\}, t = 2, \ldots, T$. We have $S = K_T - K_{T-1}$ sequences $(\mathbf{c}_{k_t}, \mathbf{A}_{k_t}, \mathbf{B}_{k_t}, \mathbf{b}_{k_t}, \mathbf{l}_{k_t}, \mathbf{u}_{k_t}), t = 2, \ldots, T$ of possible realizations of coefficients in the objective function, in recourse matrices $\mathbf{A}_*$, transition matrices $\mathbf{B}_*$, right-hand sides and bounds in the constraints for all stages. The *path probabilities* $p_{k_t} > 0$ for $\forall k_t$, $\sum_{k_t \in \mathcal{K}_t} p_{k_t} = 1$, $t = 1, \ldots, T$, of partial sequences of coefficients are probabilities of realizations of $\omega^{t-1,\bullet}$ for $\forall t$. We can obtain the path probabilities by stepwise multiplication of the marginal probabilities p_{k_2} by the conditional *arc* or *transition* probabilities $\pi_{k_\tau, k_{\tau+1}}, \tau = 2, \ldots, t$ which are related to the corresponding partial sequences of realizations. Probabilities p^k of individual scenarios

$\omega^k, k = 1, \ldots, K$, are equal to the corresponding path probabilities $p_{k_T}, k_T \in \mathcal{K}_T$. We define sets of *descendants* of k_t which consists of these indices $k_{t+1} \in \mathcal{K}_{t+1}$ for which the transition probability $\pi_{k_t,k_{t+1}} \neq 0$, we denote the sets by $\mathcal{D}esc(k_t)$.

Problem (4.1.3) may correspond to a T-period two-stage stochastic program based on the same scenarios. As it is in [10], we suppose that except for the root, there is only one descendant $d(k_t)$ of each of the t^{th}-stage nodes, that means that the transition probabilities $\pi_{k_t,d(k_t)} = 1$ for $\forall k_t \in \mathcal{K}_t, t = 2, \ldots, T-1$. Scenarios are then identified by sequences $\{k_2, \ldots, k_T\}$ such that $k_t \in \mathcal{K}_T, k_{t+1} = d(k_t)$ for $\forall t = 2, \ldots, T$. Replacing the objective function by

$$\mathbf{c}_1^T \mathbf{x}_1 \; + \sum_{k_T \in \mathcal{K}_T} p_{k_T} \left[\mathbf{c}_{k_2}^T \mathbf{x}_{k_2} \; + \; \mathbf{c}_{k_3}^T \mathbf{x}_{k_3} \; + \; \ldots \; + \mathbf{c}_{k_T}^T \mathbf{x}_{k_T} \right],$$

we obtain the *two-stage relaxation* of the multistage stochastic linear program (4.1.3).

4.2 Stages and Horizon

For applications it is crucial to build a reasonable model of the problem and to generate meaningful scenarios and this has become the most demanding task. It is important to formulate the goals and constraints and to identify the driving random process ω. A scenario-based multistage stochastic program requires specification of the horizon, stages and generation of the input in the form of a scenario tree. Practically, according to [10] and [14], we can distinguish various situations:

- Horizon and stages are determined ad hoc. This case often occurs for the purpose of testing numerical approaches and/or software.
- The horizon and stages are determined, mostly by the real-life technological process.
- The horizon depends on a fixed date (e.g. end of the fiscal year, date related with the annual Board of Directors' meeting, etc.), whereas stages are sometimes determined by the nature of the solved problem (e.g. expiration dates of options, periodic management review meetings, etc). For other cases, application of heuristic rules and/or experience, taking into account limitations due to numerical tractability, is used. In that case, rolling forward after the T-stage has been solved, the first-stage decision accepted and new information exploited, means to solve a subsequent $T-1$-stage stochastic program with a reduced number of stages or another T-stage problem with a different topology of stages.
- The horizon is connected with a time interval of a fixed (could be even infinite) length, which might be given by the periodicity of the underlying random process. The number of stages is chosen in dependence on the available computing facilities. Here, rolling forward means repeated solution of T-stage problem of the same structure of stages. The initial state of the

system is determined by the applied first-stage decision and by observation of the value ω_1. We use process ω shifted in time.

The multistage problem can be solved just once (e.g. retiring a debt by a given deadline as much as possible) or the problem and its solution goes on in the future, with new horizons, considering always just the final state of the system at the previous termination date, i.e. at the previous horizon. To ensure such continuation, the models should be usually extended by additional constraints and/or terms in the objective function to reduce the end effects, with or without reference to an additional, auxiliary stage.

The crucial step is to relate the time instants and stages for a chosen horizon. The main limitations of the number of stages occur because of the numerical tractability. In financial applications, it is common to accept unequal lengths of time periods between subsequent stages, starting with a short time first-period. Together with the repeated rolling of the model over time, this might replace quite well the full dynamics of the decision process (even for problems with a few stages). Another way of dealing with that is to break the problem with a long (or infinite) horizon into three phases: use the scenario tree structure for $1 \leq t \leq T$; design just one descendant for each node for $T+1 \leq t \leq \tau$; aggregate the rest of the process into one additional stationary stage. In case of using the previous three-phase structure, we approximate the true probability of ω by a simpler one for the reason of numerical tractability.

4.3 General Description of Scenarios and Scenario Trees

We still assume, that the probability distribution P of ω is discrete and concentrated on a finite number of points, $\omega^1, \ldots, \omega^K$. The condition on a sensible scenario at stage t is

$$\omega_\tau \in \mathcal{S}_\tau(\omega^{\tau-1,\bullet}) \qquad \text{for} \quad \forall \tau > 1.$$

Therefore the set of all considered scenarios is

$$\mathcal{S} := \left\{\omega^1, \ldots, \omega^K\right\} = \left\{\omega \mid \omega_t \in \mathcal{S}_t(\omega^{t-1,\bullet}) \quad \text{for} \quad \forall t > 1\right\}.$$

The sets of all different scenarios $\omega^{t,\bullet}$ at stage t which satisfy $\mathcal{S} := \mathcal{S}^T$ is denoted by $\mathcal{S}^t$. We will denote by $\mathcal{S}_t$ the supports of the marginal probability distributions of ω_t, $t = 1, \ldots, T$. In that concept, we call the conditional probabilities $P(\omega_t \mid \omega^{t-1,\bullet})$ on $\mathcal{S}_t(\omega^{t-1,\bullet})$ for $t > 1$ and the marginal probabilities $P(\omega_1)$ on $\mathcal{S}_1$ *arc probabilities*. We call their products $P(\omega^{t-1,\bullet}) = P(\omega_1) \prod_{\tau=2}^{t-1} P(\omega_\tau \mid \omega^{\tau-1,\bullet})$ *path probabilities*. The probability of scenario $\omega^k = \{\omega_1^k, \ldots, \omega_T^k) \in \mathcal{S}$ is p_k and it is computed as

$$p_k = P(\omega^k) = P(\omega_1^k) \prod_{\tau=2}^{T} P(\omega_\tau^k \mid \omega_1^k, \ldots, \omega_{\tau-1}^k).$$

For the scenario-based multistage stochastic programs as in (4.1.3) a special structure of the input is required. We can suppose an oriented graph, which starts from an only one node at level 0 (the root) and branches into nodes at level 1, where each of the nodes corresponds to one of the possible realizations of ω_1. The branching then continues up to nodes at level $T-1$, which are assigned to the whole possible data paths $\omega^{T-1,\bullet}$. If ω_T contributes to the overall costs, the branching continues up to T-level and level T is assigned to the whole possible data paths $\omega^{T,\bullet}$. There is one-to-one correspondence between the sections $\omega^{t,\bullet}$ and the nodes of the tree at stage t for $t = 1, \ldots, T$. We can represent this structure of the input data as a scenario tree. It follows from the one-to-one correspondence that for any node at level t, each of the new observations ω_t must have only one immediate predecessor $\omega^{t-1,\bullet}$ (ancestor, a node at level $t-1$) and it has a (finite) number of descendants ω_{t+1}, which are at nodes at level $t+1$, $t < T-1$. The number of descendants of all nodes at a given level $0 \leq t < T-1$ of the scenario tree can be equal. If this holds for all stages, we have a *balanced scenario tree*. If the scenario tree is balanced, it can be coded as a product of numbers of descendants of the root and of nodes at levels $1, \ldots, T-1$. A scenario tree with 2 branches from the root, 3 branches from nodes at the first level and no branching at the second level is described by $2^1 3^1 1^1$. This tree is depicted on the following Figure F4.3.1.

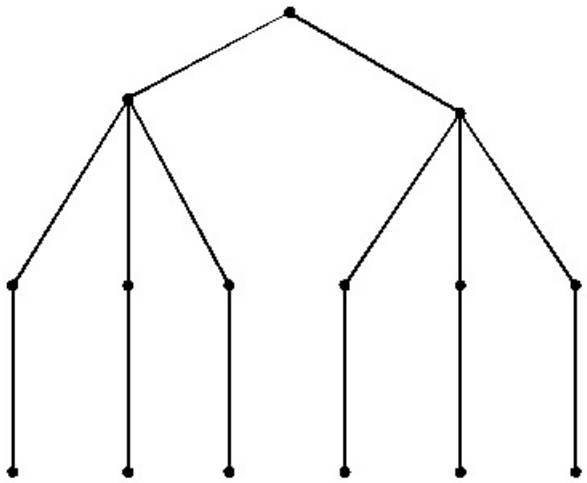


Figure F.4.3.1. Example of a tree $2^1 3^1 1^1$.

There are two special cases of the scenario tree that should be mentioned:

- For all stages $t = 2, \ldots, T-1$, the conditional probabilities $P(\omega_t \mid \omega^{t-1,\bullet})$ are independent on $\omega^{t-1,\bullet}$, they are equal to the marginal probabilities $P(\omega_t)$. We call this case *interstage independence*.

- The supports $\mathcal{S}_t(\omega^{t-1,\bullet})$ of conditional probability distributions of ω_t conditioned by realizations $\omega^{t-1,\bullet} = \{\omega_1, \ldots, \omega_{t-1}\}$ of sections $\omega^{t-1,\bullet}$ are singletons for all stages $t = 2, \ldots, T-1$. The scenario tree is then a *fan* of individual scenarios $\omega^k = \{\omega_1^k, \ldots, \omega_{T-1}^k\}$, whose probabilities are $p_k = P(\omega_1^k)$ for

$\forall k$. The multiperiod stochastic program reduces to the two-stage stochastic program, independently of the number of periods.

In applications the crucial problem is to build a representative scenario tree (except for the two special cases mentioned above). We can approach that from the point of view of a suitable data manipulation, which should reflect both the underlying probability assumptions, the existing data and it should be linked with the purpose of the application. We should strike a balance between a manageable problem size and the desired precision of the results. For that task, the important problems are designing strategies for aggregating nodes and stages, trimming or refining trees, testing the influence of including additional scenarios and stages. Number of nodes of a scenario tree grows exponentially with the number of stages, therefore an alternative data arrangement might be considered. This data arrangement could be obtained for example by relaxation of the requirement of unique predecessors at the previous stages. The following Example E4.3.1. inspired by [42] will show the general description of scenario trees.

Example E4.3.1. We will consider the scenario tree in Figure F4.3.2. Numbers along the arcs represent the arc probabilities (probabilities of moving from one node to the next.). As it was in case of linear multistage stochastic programs (subsection 3.2.2), we will denote by $\boldsymbol{\xi}_1 := (\mathbf{c}_1, \mathbf{A}_1, \mathbf{b}_1)$ the vector of known first-stage coefficients. Then for the $t = 2, \ldots, T$, the vector of random coefficients of the t^{th}-stage the vector of random coefficients will be denoted by $\boldsymbol{\xi}_t := [\mathbf{c}_t(\omega_{t-1}), \mathbf{B}_t(\omega_{t-1}), \mathbf{A}_t(\omega_{t-1}), \mathbf{b}_t(\omega_{t-1})]$. Here we have T = 4. We assume here that all involved variables are one dimensional, where c_t, B_t, A_t are fixed for $t = 2, 3, 4$ and only $b_t, t = 2, 3, 4$ are random. The realizations of the random process $b_1, b_2(\omega_1), b_3(\omega_2), b_4(\omega_3)$ are indicated by the numbers in the nodes of the tree. Here we have that at level 0 ($t = 1$), b_1 has a unique value 100. At level 1 ($t = 2$) we have that b_2 has three values $90, 61, 110$ with probabilities $0.3, 0.35, 0.35$ respectively. At level 2 ($t = 3$), we have 6 nodes, from left to right $60, 93, 91, 105, 101, 60$. Therefore b_3 takes 5 different values with the respective probabilities $P[b_3 = 60] = 0.3 \times 0.3 + 0.35 \times 0.8, P[b_3 = 93] = 0.3 \times 0.70, P[b_3 = 91] = 0.35 \times 0.4, P[b_3 = 105] = 0.35 \times 0.6$ and$P[b_3 = 101] = 0.35 \times 0.2$. At level 3 ($t = 4$), the numerical values associated with 10 nodes are (from left to right) $50, 70, 70, 80, 85, 50, 95, 92, 50, 115$. The respective probabilities can be calculated as for b_3, e.g. $P[b_4 = 50] = 0.3 \times 0.3 \times 0.5 + 0.35 \times 0.6 \times 0.75 + 0.35 \times 0.8 \times 0.7$. It is important to mention, that although some of the realizations of b_3 (and hence $\boldsymbol{\xi}_3$) are equal to each other, we represent them by different nodes, because we identify different histories of the process corresponding to different scenarios. The same reason applies for b_4 (and hence $\boldsymbol{\xi}_4$). There are 10 scenarios in this tree.

We can mention that the process b_t (and hence $\boldsymbol{\xi}_t$) in the example is not Markovian. Take the example of

$$P[b_4 = 50 \mid b_1 = 100, b_2 = 110, b_3 = 60] = 0.70,$$

while

$$P[b_4 = 50 \mid b_3 = 60] = \frac{P[b_4 = 50, b_3 = 60]}{P[b_3 = 60]}$$

$$= \frac{0.3 \times 0.3 \times 0.5 + 0.35 \times 0.8 \times 0.7}{0.3 \times 0.3 + 0.35 \times 0.8} \doteq 0.65 \neq 0.70.$$

The process b_t in this example is also not a martingale[4], for example

$$\mathbf{E}[b_2 \mid b_1 = 100] = \mathbf{E}[b_2] = 90 \times 0.3 \ + \ 61 \times 0.35 \ + \ 110 \times 0.35 = 86.85$$
$$\mathbf{E}[b_3 \mid b_2 = 90, b_1 = 100] = 60 \times 0.3 \ + \ 93 \times 0.7 = 83.1, \text{ etc.}$$

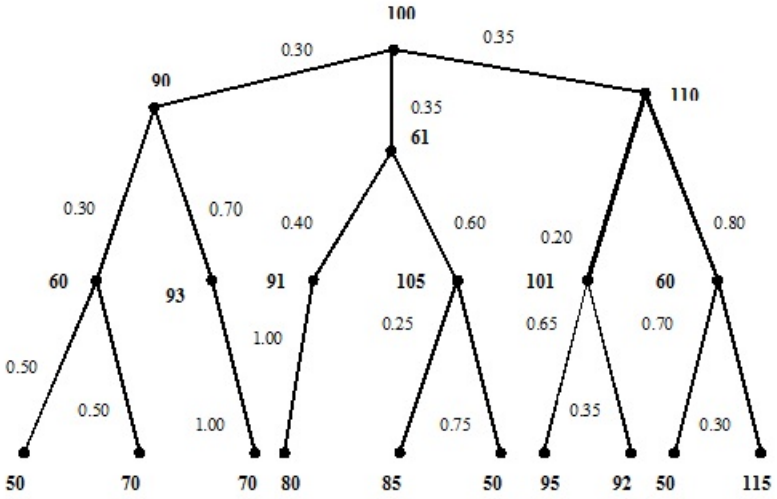


Figure F4.3.2 Considered scenario tree.

[4]We call a random process $\{Z_t, t \in \mathcal{N}\}$ a martingale, if the inequalities $\mathbf{E}[Z_{t+1} \mid Z_{[1,t]}] = Z_t, t \in \mathcal{N}$ hold with probability one.

5. Scenarios and Distances

The typical assumption in the general formulation of the stochastic programming problems is that the probability distribution P of the stochastic element ω is known. However, this seems to be not a very realistic assumption. We usually try to approximate the true probability distribution P. To get a reasonable approximation, we should exploit the structure of the problem, the available information about P that comes from theory, historical data, experts' experience, heuristics etc. If we reduce the approximation of P to the approximation of one-dimensional probability distributions, there are several ways of doing this, e.g. approximation by a discrete probability distribution, piecewise uniform distributions, kernel estimates etc. In the multidimensional case, the approximation by discrete probability distributions is usually used. We remind that the true probability distribution P is replaced by a *discrete* probability distribution $\hat{P}$, which is concentrated on a finite number of points (scenarios), say $\omega^1, \ldots, \omega^K$, with respective probabilities $p^1, \ldots, p^K$. The aim of scenario generation is to represent the true probability distribution$P$ in a reasonable way. We often need to compromise between the precision of the approximation and the size of the approximated problem, we also often need a special form of the input - scenario tree for multistage stochastic programming problems. The origin of scenarios might be diverse - they can be atoms of a known genuine discrete probability distribution, they could be obtained in the course of a discretization approximation scheme, they could be obtained by simulation or by limited sample information. They can also be the result of recognized regulations or of a preliminary analysis of the problem with probabilities of their occurence (the probabilities may reflect an ad hoc belief, subjective opinion of an expert, etc.).

5.1 Scenarios and Their Generation

According to [14] and [11] four basic types of problems can be distinguished concerning the level of the available information.

1) Full knowledge of the probability distribution
The probability distribution P is fully specified. In that case, we can obtain the scenarios by sampling from this distribution or application of a simulation or discretization scheme. There are several possibilities how the conclusions about the optimal solution of the original problem can be drawn according to the chosen approximation technique. The assumed full knowledge of the probability distribution may originate from a theoretical model, from historical data or from an experience of an expert.

2) Known parametric family
The given parametric form of the distribution P is preferably based on a theoretical model and the parameters of the probability distribution P are estimated from the available data. The parametric form of the probability distribution or of

the stochastic process is chosen correspondingly to the choice of the model. The estimation of parameters corresponds to the calibration of the model and then, as in the case of known probability distribution, follows simulation, sampling or discretization procedure. This case usually occurs in stochastic programming problems in finance and water resources management and planning. This is partly because of the fact that the relevant stochastic models of interest rates and asset prices or those of water inflows have been studied for a relatively long time and they have been well developed and supported by historical data. These days, the interest is to build appropriate models for distributions of demand. As an example, we can mention *The Vector Autoregressive Models, Vašíček's Model for Spot Rates* and *Multidimensional, Multifactor Models*, all described in [14]. *The Black-Derman-Toy Model* introduced in [3] is one of frequently used discrete time models for generation of interest rate scenarios. We should also mention the *Multivariate GARCH model* proposed in [1].

3) Sample information
In that case, the available sample information about the true probability distribution is based mostly on observed past data. For enough homogeneous data, we can view them as independent, identically distributed (i.i.d.) random variables (vectors), hence the use of empirical distributions is straightforward. If the data are not homogeneous enough, we could think of a preprocessing procedure to treat the missing data, smoothing, etc., or of an adjustment to fit specific values of (sample) moments. One simple idea is to use as scenarios the past observations which were obtained under comparable circumstances and assign them equal probabilities (*distribution- free* method).

4) Low information level
In case we have no reliable data, the previously mentioned procedures fail. Then the scenarios and their probabilities are mostly based on experts' forecasts or even governmental regulations. Sometimes, it is also the case, when the true probability distribution is described only by several moment values or/and some simple qualitative properties.

Example E5.1.1. Vector Autoregressive models. This model is described in [14] and it is an example of the discrete time stochastic models. The vector autoregressive model of the first order for our stochastic (vector) process $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_T)$ is

$$\boldsymbol{\omega}_t \quad = \quad \boldsymbol{\mu} \quad + \quad \mathbf{H}(\boldsymbol{\omega}_{t-1} - \boldsymbol{\mu}) \quad + \quad \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{\Sigma}), \tag{5.1.1.}$$

where the eigenvalues of the matrix $\mathbf{H}$ fulfill the condition $|\lambda(\mathbf{H})| < 1$ and $\boldsymbol{\varepsilon}_t$'s are jointly independent. We estimate the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{H}$ from historical data and we possibly further adapt it to distinct sources of information, such as experts' opinions, forecasts, related global parameters, etc. We will denote these estimates by $\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}, \hat{\mathbf{H}}$.

We construct scenarios in the following way. We start from a known vector $\boldsymbol{\omega}_0$ and we use the calibrated model (5.1.1.), then step by step we construct the scenarios as

$$\boldsymbol{\omega}_t^k \quad = \quad \hat{\boldsymbol{\mu}} \quad + \quad \hat{\mathbf{H}}(\boldsymbol{\omega}_{t-1}^k - \hat{\boldsymbol{\mu}}) \quad + \quad \hat{\boldsymbol{\varepsilon}}_t^k,$$

where $\hat{\boldsymbol{\varepsilon}}_t^k$ is obtained as an observation from $N(\mathbf{0}, \hat{\boldsymbol{\Sigma}})$ by a suitable discretization or simulation technique.

5.2 Approximation of The True Problem

If we want to make a conclusion about the optimal solutions and the optimal value of the true stochastic program, while using the results of the approximated scenario-based problem, we have to keep in mind, that they depend on the structure of the solved problem as well as on the origin of scenarios. The output can hardly be more precise than the input. It is much easier to think about the precision of the obtained optimal values than the sets of optimal solutions. We will consider a stochastic program in the form:

$$\text{minimize} \quad F(\mathbf{x}, P) := \mathbf{E} f_0(\mathbf{x}, \omega) \tag{5.2.1}$$

on a closed nonempty set $\mathcal{X} \subset \mathbb{R}^n$, which is independent of P. The optimal value of the problem (5.2.1) will be denoted by $\varphi(P)$, the set of optimal solutions (not necessarily a singleton) by $\mathcal{X}^*(P) = \arg\min_{\mathbf{x}\in\mathcal{X}} f_0(\mathbf{x}, P)$. If $\mathcal{X}^*(P)$ is a singleton, then the unique optimal solution of (5.2.1.) is denoted by $\mathbf{x}^*(P)$. We assume that the true probability distribution P has been replaced by another probability distribution $\hat{P}$, which was obtained by parametric or nonparametric methods and by sampling, discretization and simulation techniques. We can quantify the precision of the approximation by a suitable measure of distance between these two probability distributions. This approach will be described in the next section. Here, the qualitative results, as they are in [14], will be discussed.

Definition 9. Let P, $P_\tau, \tau = 1, \ldots,$ be probability measures on Borel sets of the same Euclidean space Ω. We say, that P_τ *converges weakly to* P as $\tau \to +\infty$, if for any bounded continuous function $h : \Omega \to \mathbb{R}^1$ holds:

$$\int_\Omega h(\omega)\, P_\tau(d\omega) \quad \longrightarrow \quad \int_\Omega h(\omega)\, P(d\omega).$$

We will restrict the class of the considered functions $f_0(\mathbf{x}, \cdot)$ to bounded continuous functions of ω or restrict the set of probability measures to a subset with respect to which the functions $f_0(\mathbf{x}, \cdot)$ are uniformly integrable to obtain continuity of expectation functionals $F(\mathbf{x}, P) := \int_\Omega f_0(\mathbf{x}, \omega)\, P(d\omega)$. We will assume that $f_0(\mathbf{x}, \omega)$ is a continuous bounded function of ω for every $\mathbf{x} \in \mathcal{X}$ and $P_\tau \to P$ weakly. Definition 9 gives us then the pointwise convergence of the objective functions in (5.2.1),

$$F(\mathbf{x}, P_\tau) \to F(\mathbf{x}, P) \qquad \text{for} \quad \forall\, \mathbf{x} \in \mathcal{X}.$$

For $\mathcal{X}$ compact and uniform convergence on $\mathcal{X}$, we obtain for the optimal values:

$$\varphi(P_\tau) \to \varphi(P).$$

Furthermore, for $\mathcal{X}$ convex and $f_0(\cdot,\omega)$ strictly convex on $\mathcal{X}$ for $\forall\,\mathbf{x}$, we have the convergence of the unique optimal solutions $\mathbf{x}^*(P_\tau)$ of $\min\limits_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x},P_\tau)$ to the unique optimal solution $\mathbf{x}^*(P)$ of the problem (5.2.1).

Results about the bias for the case of the empirical optimal value are shown in [10]. We will suppose, that P_τ are empirical probability distributions based on τ i.i.d. sample values of ω. For empirical probability distributions, $\varphi(P_\tau)$ has an one-directional bias,

$$\mathbf{E}\varphi(P_\tau) \leq \varphi(P).$$

For ω^i being i.i.d. and for any fixed $\mathbf{x} \in \mathcal{X}$, we have that the function values $f_0(\mathbf{x},\omega^j)$ are i.i.d. and

$$\mathbf{E}\varphi(P_\tau) = \mathbf{E}\min_{\mathbf{x}} \tfrac{1}{\tau}\sum_{j=1}^{\tau} f_0(\mathbf{x},\omega^j) \leq \min_{\mathbf{x}} \tfrac{1}{\tau}\mathbf{E}\sum_{j=1}^{\tau} f_0(\mathbf{x},\omega^j) =$$

$$= \min_{\mathbf{x}} \mathbf{E} f_0(\mathbf{x},\omega) = \varphi(P).$$

The empirical point estimate of $\mathbf{E}\varphi(P_\tau)$ can be obtained from the Law of Large Numbers and the asymptotic confidence interval for $\varphi(P)$ can be constructed from the Central Limit Theorem.

We will assume that we know that the true probability distribution P belongs to a parametric family $\mathcal{P} = \{P_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ of probability distributions, where the parameter vector $\boldsymbol{\theta}$ belongs to an open set $\boldsymbol{\Theta} \subset \mathbb{R}^q$. In that case, the objective function depends on the parameter $\boldsymbol{\theta}$, $(F(\mathbf{x},P_{\boldsymbol{\theta}}) := F(\mathbf{x},\boldsymbol{\theta}))$, therefore the problem (5.2.1) is a standard parametric program in the form $\min\limits_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x},\boldsymbol{\theta})$. Suppose for $\forall\,\boldsymbol{\theta} \in \boldsymbol{\Theta}$ the optimal value $\varphi(\boldsymbol{\theta})$ exists and it is a continuous function of $\boldsymbol{\theta}$ on a neighborhood of the true parameter value $\boldsymbol{\theta}_0$. Supposing we have a statistical estimate $\boldsymbol{\theta}_\tau$ of the true parameter $\boldsymbol{\theta}_0$, we have, that whenever $\boldsymbol{\theta}_\tau \to \boldsymbol{\theta}_0$ with probability 1 or in probability, then $\varphi(\boldsymbol{\theta}_\tau) \to \varphi(\boldsymbol{\theta}_0)$ with probability 1 or in probability, respectively.

For $\boldsymbol{\theta}_\tau$ an asymptotically normal estimate of $\boldsymbol{\theta}$ (e.g. $\sqrt{\tau}(\boldsymbol{\theta}_\tau - \boldsymbol{\theta}) \sim N(\mathbf{0},\boldsymbol{\Sigma})$) and φ continuously differentiable at $\boldsymbol{\theta}$ with $\nabla\varphi(\boldsymbol{\theta}) \neq 0$, we have that $\varphi(\boldsymbol{\theta}_\tau)$ is asymptotically normal,

$$\sqrt{\tau}(\varphi(\boldsymbol{\theta}_\theta) - \varphi(\boldsymbol{\theta})) \sim N(\mathbf{0}, \nabla\varphi(\boldsymbol{\theta})^T\boldsymbol{\Sigma}\nabla\varphi(\boldsymbol{\theta})).$$

5.2.1 The Contamination Method

This method described in [12], [9], [10] is initiated in mathematical statistics as one of the tools for analysis of robustness of estimators with respect to deviations from the assumed probability distribution and/or its parameters. It does not require any specific properties of the probability distribution P. The robustness analysis with respect to changes in P is reduced to a much simple analysis with

respect to a scalar parameter λ. We still assume the stochastic program in the form (5.2.1). This method is suitable for postoptimality analysis (a technique for determining how the optimal solution changes if the problem data change), it may be used to support conclusions about resistance of the already obtained optimal output to changes of scenarios and their probabilities and also to check possible influence of out-of-sample scenarios. We assume that the problem (5.2.1.) has been solved for an already constructed scenario tree, which corresponds to the discrete probability distribution P. The set of optimal values is $\varphi(P)$ and the set of optimal solutions is $\mathcal{X}^*(P)$.

We model the changes in the probability distribution (e.g. because of including additional scenario branches of the scenario tree) using contamination distributions P_λ as follows:

$$P_\lambda = (1-\lambda)P \,+\, \lambda Q, \qquad 0 \leq \lambda \leq 1. \tag{5.2.1.1}$$

The probability distribution Q (e.g. the probability distribution of the additional scenarios or branches of the scenario tree) is fixed and it contaminates the original probability distribution P. Suppose the probability distributions P, Q are fixed. If we compute the expected value in (5.2.1.) for the contaminated distribution P_λ for an objective function in (5.2.1.) linear in P, it is *linear* in the parameter λ, e.g.

$$F(\mathbf{x}, \lambda) := \int_\Omega f_0(\mathbf{x}, \omega)\, P_\lambda(d\omega) \,=\, (1-\lambda)F(\mathbf{x}, P) \,+\, \lambda F(\mathbf{x}, Q).$$

We suppose that the stochastic program (5.2.1.) has an optimal solution for all distributions P_λ, $0 \leq \lambda \leq 1$. The optimal value function of problem (5.2.1)

$$\varphi(\lambda) := \min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}, P_\lambda)$$

is a finite concave function on $[0,1]$, hence it is continuous and the directional derivatives exist on $(0,1)$. Let $\mathcal{X}^*(P)$ be a nonempty, bounded set of optimal solutions of the stochastic program (5.2.1.). Using the stationarity of the derivatives $\frac{dF(\mathbf{X},\lambda)}{d\lambda} = F(\mathbf{x}, Q) - F(\mathbf{x}, P)$, the form of the directional derivatives is

$$\varphi^{'}(0^+) = \min_{\mathbf{x}\in\mathcal{X}^*(P)} F(\mathbf{x}, Q) - \varphi(0).$$

The bounds of the optimal value function $\varphi(\lambda)$ are

$$(1-\lambda)\varphi(0) \,+\, \lambda\varphi(1) \leq \varphi(\lambda) \leq \varphi(0) \,+\, \lambda\varphi^{'}(0^+), \qquad \lambda \in [0,1].$$

We construct the right upper bound by interchanging the role of probability distributions P, Q.

Suppose that $\mathbf{x}^*(P)$ is the unique solution of (5.2.1.), $\varphi^{'}(0^+) = F(\mathbf{x}^*(P), Q) - \varphi(0)$. The local change of the optimal value function caused by a small change of P in direction $Q-P$ is the same as that of the objective function at $\mathbf{x}^*(P)$. In case of multiple optimal solutions, each of them leads to an upper bound $\varphi^{'}(0+) \leq F(\mathbf{x}(P), Q) - \varphi(0), \mathbf{x}(P) \in \mathcal{X}^*(P)$. Then we can write the contamination bounds as

$$(1-\lambda)\varphi(P)+\lambda\varphi(Q)\leq\varphi(P_\lambda)\leq(1-\lambda)\varphi(P)+\lambda F(\mathbf{x}(P),Q) \tag{5.2.1.2.}$$

for an arbitrary $\mathbf{x}(P)\in\mathcal{X}^*(P)$ and $\lambda\in[0,1]$. For $\mathbf{x}(\mathrm{P})$ being an ε-optimal solution of (5.2.1.) for probability distribution Q, the difference of the upper and lower bound in (5.2.1.2.) is less or equal to $\lambda\varepsilon$.

If we consider only small changes in the underlying probability distribution, we typically use small values of the contamination parameter λ, e.g. when incorporating expert opinions represented by the contaminating distribution Q, the choice of λ might reflect, how confidential this opinion is. Sometimes, we use the contaminating distribution Q of additional scenarios or branches of the scenario tree to study the influence of including these *out-of-sample* scenarios. This method can be also applied when we are interested in quantization of the response on an increasing importance of a scenario for an appropriate choice of Q. Contamination method for multistage stochastic linear programs is shown in [9].

5.2.2 The Minimax Approach

When the probability distribution P is not completely known, the worst-case analysis and the minimax decision rule are usually used [10]. We assume that P belongs to a known family $\mathcal{P}$ of probability distributions, which is identified for instance by a given support, by known values of some moments, by qualitative properties, etc. Applying the minimax decision rule to the problem in the form (5.2.1.) means to select the decision $\mathbf{x}^*$, such that the largest possible expectation is minimized:

$$\mathbf{x}^*\in\arg\min_{\mathbf{x}\in\mathcal{X}}\max_{P\in\mathcal{P}}F(\mathbf{x},P). \tag{5.2.2.1.}$$

The objective function of the inner maximization problem in (5.2.2.1.) is linear in P, therefore for convex, compact set $\mathcal{P}$, the optimal *worst-case* probability distribution $P^*\in\mathcal{P}$ is one of the extremal points of $\mathcal{P}$. These results are useful for construction of the minimum and maximum bounds of the optimal value of the true program, which are valid for all possible distributions P belonging to $\mathcal{P}$,

$$\min_{\mathbf{x}\in\mathcal{X}}\inf_{P\in\mathcal{P}}F(\mathbf{x},P)\leq\varphi(P)\leq\min_{\mathbf{x}\in\mathcal{X}}\sup_{P\in\mathcal{P}}F(\mathbf{x},P).$$

This approach gives us the best-case and the worst-case discrete probability distributions carried by fully specified scenarios for some special assumptions about the family $\mathcal{P}$ and about the structure of the underlying stochastic program (5.2.1.), such as convexity or concavity of the random objective function $f_0(\mathbf{x},\omega)$ and/or separability with respect to components of ω.

5.3 Quantitative Aspects of The Approximation

We assume here the stochastic optimization problem

$$\min\left\{\int f(\mathbf{x},\omega)\,dP(\omega) : \mathbf{x}\in\mathcal{X}\right\} \tag{5.3.1.}$$

,

where P can be a continuous or discrete. Our aim still remains to find a simple distribution $\hat{P}$ with only a few mass points, such that the original problem (5.3.1.) can be well approximated by the problem

$$\min\left\{\int f(\mathbf{x},\omega)\,d\hat{P}(\omega) : \mathbf{x}\in\mathcal{X}\right\}. \tag{5.3.2.}$$

As it is in [22], we would like to find the approximation, such that the difference between the objective functions of the two problems,

$$\sup\left\{\left|\int f(\mathbf{x},\omega)\,dP(\omega) \;-\; \int f(\mathbf{x},\omega)\,d\hat{P}(\omega)\right| : \mathbf{x}\in\mathcal{X}\right\}, \tag{5.3.3.}$$

is small. It can be regarded as finding the appropriate distance d, such that minimizing $d(P,\hat{P})$ leads to a small value of the difference (5.3.3.).We can achieve optimal approximations of stochastic programs in the sense of the difference in (5.3.3.) by minimizing some probability metric. Let $\mathcal{H}$ be a class of measurable functions. Typically the distances of distributions are of the form

$$d_{\mathcal{H}}(P,\hat{P}) = \sup\left\{\left|\int f(\omega)dP(\omega) \;-\; \int f(\omega)d\hat{P}(\omega)\right| : f\in\mathcal{H}\right\}.$$

We say, that $\mathcal{H}$ is separating, if $d_{\mathcal{H}}(P,\hat{P})=0$ implies that $P=\hat{P}$. For that case we call d a distance. For $\mathcal{H}$ containing the functions $\{f(\mathbf{x},\cdot) : \mathbf{x}\in\mathcal{X}\}$, we have that

$$\left|\int f(\mathbf{x},\omega)\,dP(\omega) \;-\; \int f(\mathbf{x},\omega)\,d\hat{P}(\omega)\right| \le d_{\mathcal{H}}(P,\hat{P}).$$

One important function class for approximation of stochastic programs are *Lipschitz continuous functions*. For $p=0$ and $p\ge 0$ and $\Omega\subset\mathbb{R}^k$, class $\mathcal{H}_p(\Omega)$ of Lipschitz continuous functions of order p is described as:

$$\mathcal{H}_0(\Omega) = \mathcal{H}_1(\Omega)\cap\left\{f\in C_b(\Omega) : \sup_{\omega\in\Omega}|f(\omega)|\le 1\right\},$$

$$\mathcal{H}_p(\Omega) = \left\{f:\Omega\to\mathbb{R} : |f(\omega)-f(\tilde{\omega})|\le \max(1,||\omega||,||\tilde{\omega}||)^{p-1}||\omega-\tilde{\omega}||,\ \forall\,\omega,\tilde{\omega}\in\Omega\right\}.$$

According to the value of p, we call the distance

$$d(P,\hat{P}) = \sup\left\{\left|\int_{\Omega} f(\omega)dP(\omega) \;-\; \int_{\Omega} f(\omega)d\hat{P}\right| : f\in\mathcal{H}_p(\Omega)\right\},$$

- *Bounded Lipschitz metric* d_{BL} for $p = 0$,
- *Wasserstein (Kantorovich) metric* d_W for $p = 1$,
- *Fortet-Mourier metric* d_{FM_p} of order p for $1 < p < +\infty$,
- *Kolmogorov (Uniform) metric* d_K for $p = +\infty$.

The Wasserstein metric d_W is related to the mass transportation problem. The properties of the Wasserstein distance are given in the following theorem stated in [35], the proof (i) is to be found in [50] and (ii) in [40], (iii) is a consequence of (ii).

Theorem 16. *The properties of* d_W *are*

(i) *Kantorovich - Rubinstein*
$d_W(P, \hat{P}) = \inf\{\mathbf{E}(|X - \hat{X}|;$ *where the joint distribution*$(X, \hat{X})$ *is arbitrary, but the marginal distributions are fixed such that* $X \sim P;\ \hat{X} \sim \hat{P}\}$,

(ii) *for one-dimensional distributions,* d_W *is defined as*

$$d_W(P, \hat{P}) = \int_\Omega |P(\omega) - \hat{P}(\omega)| = \int_\Omega |P^{-1}(\omega) - \hat{P}^{-1}(\omega)|,$$

where $P^{-1}(\omega) = \sup\{v : P(v) \leq \omega\}$,

(iii) *among all one-dimensional* $\hat{P}$, *which sit on the mass points* $z_1, z_2, \ldots, z_m$, *the one closest to* P *in* d_W*-distance has masses*

$$p_i = P\left(\frac{z_i + z_{i+1}}{2}\right) - P\left(\frac{z_i + z_{i-1}}{2}\right),$$

where $z_0 = -\infty$ *and* $z_{m+1} = +\infty$. *For this* $\hat{P}$,

$$d_W(P, \hat{P}) = \sum_{i=1}^{m} \int_{\frac{z_{i-1}+z_i}{2}}^{\frac{z_i+z_{i+1}}{2}} |\omega - z_i|\, dP(\omega),$$

the infimum in (i) *is attained.*

In [22] it is shown, that it is possible to relate the Fortet-Mourier distance to the Wasserstein distance.

We will denoted by $\mathcal{B}$ a set of all Borel subsets of Ω. We will then define the $\mathcal{B}$-discrepancy as in [22] as

$$d_D(P, \hat{P}) = \sup_{B \in \mathcal{B}} |P(B) - \hat{P}(B)|.$$

We have various types of discrepancy distances for different structures of Ω and $\mathcal{B}(\Omega)$. Especially, for $\Omega = \mathbb{R}^k$ and $\mathcal{B}(\Omega) = \{(-\infty, \omega] : \omega \in \mathbb{R}^k\}$, we have the Kolmogorov distance.

5.4 Distance between Stochastic Scenario Processes and Scenario Trees

This topic is discussed in [49] and some of the ideas from this paper will be presented here. We consider a stochastic process $\omega = (\omega_1, \ldots, \omega_T)$ and we will denote the σ-algebra generated by the random variable ω_t by $\sigma(\omega_t)$. Here, the filtration generated by the process ω is denoted by $\sigma(\omega) = (\sigma(\omega_1), \sigma(\omega_1, \omega_2), \ldots, \sigma(\omega_1, \omega_2, \ldots, \omega_T))$. Here, we also assume the nonanticipativity property of the process. We will define the *tree process* as in [36].

Definition 10. We call a stochastic process $\nu = (\nu_1, \ldots, \nu_T)$ a *tree process*, if $\sigma(\nu_1), \sigma(\nu_2), \ldots, \sigma(\nu_T)$ is a filtration.

If we approximate the true probability distribution of ω by a discrete one with a finite number of atoms, we call then the approximating stochastic process $\tilde{\omega}$ *finitely valued* and the tree that represents that process *finitely valued tree*. We define the bushiness of a tree as in [49]:

Definition 11. Assume that the finitely valued stochastic process $\tilde{\omega}$ is represented by a tree with the same number of successors s_t for each node at stage $t, \forall\, t = 1, \ldots, T$. The vector $bush = (s_1, \ldots, s_T)$ is a *bushiness vector* of the tree.

In general, we will consider a distance $d(P, \tilde{P})$ between two probability measures P and $\tilde{P}$. Suppose that P and $\tilde{P}$ are two Borel probability measures given on separable metric spaces (Ω, d_Ω) and $(\tilde{\Omega}, d_{\tilde{\Omega}})$ respectively. We consider a Borel probability measure π on $\Omega \times \tilde{\Omega}$ with fixed marginals $P(\cdot) = \pi(\cdot, \tilde{\Omega})$ and $\tilde{P}(\cdot) = \pi(\Omega, \cdot)$. Then we can define the Kantorovich distance for random variables from probability spaces $(\Omega, \mathcal{F}, P)$ and $(\tilde{\Omega}, \tilde{\mathcal{F}}, \tilde{P})$ as in [51]:

Definition 12. The *Kantorovich distance between random variables* is defined as

$$d_{KV}(P, \tilde{P}) = \inf_{\pi} \left\{ \int_{\Omega \times \tilde{\Omega}} d(\eta, \tilde{\eta})\, \pi[d\eta, d\tilde{\eta}] \right\},$$

where $\pi(\cdot, \tilde{\Omega}) = P(\cdot), \pi(\Omega, \cdot) = \tilde{P}(\cdot)$.

The next theorem (given in [49], proof. in references therein) gives us more insight into the discrete approximation of probability distribution P by a probability distribution $\tilde{P}^n$ sitting on n points.

Theorem 17. *Suppose that N-dimensional distribution P has a density f with $\int |u|^{1+\delta} f(u)\, du < +\infty$ for some $\delta > 0$ and suppose that*

$$\left[\int f(x_t \mid x_1, \ldots, x_{t-1})^{\frac{N}{N+1}}\, dx_t \right]^{\frac{N}{N+1}} \leq c, \quad \forall t,$$

where c is some constant.
Then

$$d(P, \tilde{P}^n) \leq cn^{-\frac{1}{N}},$$

where we denote by $\tilde{P}^n$ a discrete approximation of P sitting on n points, and for $n \to +\infty$ we have

$$d(P, \tilde{P}^n) \to 0.$$

We will now focus on considering the stochastic process together with the gradually increasing information provided by filtrations. The following definition as in [49] gives us the description of the scenario process-and-information structure:

Definition 13. Consider a tree process ν and $\mathcal{F} = \sigma(\nu)$. We can write any process ω adapted to $\mathcal{F}$ as $\omega_t = f_t(\nu_t)$ for some measurable function f_t. The pair consisting of a scenario process ω and a tree process ν, such that $\omega \lhd \sigma(\nu)$ (ω is $\sigma(\nu)$-adapted), is called a *scenario process-and-information structure* and it is denoted by $(\Omega, \mathcal{F}, P, \omega)$.

We call the distribution of the process-and-information structure a *nested distribution* $\mathbb{P}$. The relation between the nested distribution $\mathbb{P}$ and the process-and-information structure $(\Omega, \mathcal{F}, P, \omega)$ is comparable to the relation between a probability measure P on $\mathbb{R}^k$ and a $\mathbb{R}^k$-valued random variable ω with distribution P. It can be considered either the nested distribution $\mathbb{P}$ or its realization with Ω as a tree, $\mathcal{F}$ an information and ω a process. This is symbolized by $\mathbb{P} \sim (\Omega, \mathcal{F}, P, \omega)$. For defining the Kantorovich distance for multistage case, we consider two process-and-information structures, $\mathbb{P} \sim (\Omega, \mathcal{F}, P, \omega)$ and $\tilde{\mathbb{P}} \sim (\tilde{\Omega}, \tilde{\mathcal{F}}, \tilde{P}, \tilde{\omega})$. Then the definition of the multistage nested distance according to [37] is

Definition 14. The multistage distance of two process-and-information structures is

$$d(\mathbb{P}, \tilde{\mathbb{P}}) = \inf_{\pi} \left(\int_{\Omega, \tilde{\Omega}} d(\eta, \tilde{\eta})\, \pi(d\eta, d\tilde{\eta}) \right),$$

$$\begin{gathered} \mathbb{P} \sim (\Omega, \mathcal{F}, P, \omega),\ \tilde{\mathbb{P}} \sim (\tilde{\Omega}, \tilde{\mathcal{F}}, \tilde{P}, \tilde{\omega}), \\ \pi[A \times \tilde{\Omega}] \mid \mathcal{F}_t \otimes \tilde{\mathcal{F}}_t](\eta, \tilde{\eta}) = P(A \mid \mathcal{F}_t)(\eta), \quad (A \in \mathcal{F}_T, 1 \leq t \leq T), \\ \pi[\Omega \times B] \mid \mathcal{F}_t \otimes \tilde{\mathcal{F}}_t](\eta, \tilde{\eta}) = \tilde{P}(B \mid \tilde{\mathcal{F}}_t)(\tilde{\eta}), \quad (B \in \tilde{\mathcal{F}}_T, 1 \leq t \leq T), \end{gathered}$$

where $d(\eta, \tilde{\eta}) = \sum_{t=0}^{T} d_t(\omega_t(\eta_t), \tilde{\omega}_t(\tilde{\eta}_t))$ is the distance in the integral on the sample space $\Omega \times \tilde{\Omega}$ and d_t is the distance available in the state space of the processes ω_t and $\tilde{\omega}_t$, the predecessor of η which belongs to the stage t of the tree, $\forall t = 1, \ldots, T$, is denoted by η_t.

We will now consider a multistage stochastic optimization program written in the form

$$v(\mathbb{P}) = \inf\{\mathbf{E}_P[F(\mathbf{x},\omega)] : \mathbf{x} \triangleleft \mathcal{F}, \mathbf{x} \in \mathcal{X}\} = \inf\left\{\int_\Omega F(\mathbf{x},\omega)dP : \mathbf{x} \triangleleft \mathcal{F}, \mathbf{x} \in \mathcal{X}\right\}. \tag{5.4.1.1.}$$

The next theorem as in [37] gives the relationship between the nested distance and the closeness of the initial and the approximate multistage optimization program.

Theorem 18. *Assume that $\mathbb{P}$ and $\tilde{\mathbb{P}}$ are process-and information structures. Assume also that $\mathcal{X}$ is convex and the objective function F is convex in $\boldsymbol{x}$ for any ω fixed, i.e.*

$$F((1-\lambda)\boldsymbol{x}_0 + \lambda\boldsymbol{x}_1, \omega) \leq (1-\lambda)F(\boldsymbol{x}_0,\omega) + \lambda F(\boldsymbol{x}_1,\omega).$$

Furthermore, let F be uniformly Hölder continuous ($\beta \leq 1$) with constant L_β, that is

$$|F(\boldsymbol{x},\omega) - F(\boldsymbol{x},\tilde{\omega})| \leq L_\beta \left(\sum_{t\in T} d_t(\omega_t,\tilde{\omega}_t)\right)^\beta$$

for $\forall \boldsymbol{x} \in \mathcal{X}$.
Then the value function v in (5.4.1.1.) inherits the Hölder constant with respect to the multistage distance, that is $|v(\mathbb{P}) - v(\tilde{\mathbb{P}})| \leq L_\beta d(\mathbb{P},\tilde{\mathbb{P}})^\beta$.

The theorem gives us that the less distance between two process-and-information structures the closer the solutions of the initial and the approximate multistage stochastic optimization program. If we minimize the distance $d(\mathbb{P},\tilde{\mathbb{P}})$ between stochastic process given by the continuous distribution function and a tree, we will obtain the best upper bound for the difference between the true and the approximate solution $|v(\mathbb{P}) - v(\tilde{\mathbb{P}})|$.

The Kantorovich distance $d(P,\tilde{P})$ enables us to measure the stage-wise difference between continuous distribution function P and its discrete approximation $\tilde{P}$, but we are not able to measure the distance between stochastic process given by the continuous distribution function and the whole tree that contains all available information. On the other hand, we can use the multistage nested distance to measure the difference between two finite trees, let denote them by $\tilde{\mathbb{P}}^1$ and $\tilde{\mathbb{P}}^2$. We will introduce an approximation of the multistage distance between stochastic process given by its continuous distribution function and a tree with a given structure, probabilities and values as it is described in [49].

Suppose a given tree (Tree 1) with a known tree structure, probabilities and values sitting on the nodes. Assume we have a stochastic process $\omega = (\omega_1,\ldots,\omega_T)$, which is given by the joint distribution function P_J. Our aim is to find the distance between this stochastic process and the given tree. The nested distribution of the stochastic process ω is $\mathbb{P}$, the nested distribution of the given Tree1 is $\tilde{\mathbb{P}}$ and the nested distribution of the approximate stochastic process ω^* is $\mathbb{P}^*$ (Tree 2). We are aiming at calculating the distance $d(\mathbb{P},\tilde{\mathbb{P}})$ between the stochastic process ω and the given Tree 1 by the approximation of this stochastic process by a

tree with high bushiness (Tree 2). We have the triangle inequality for the nested distance, $d(\mathbb{P}, \tilde{\mathbb{P}}) \leq d(\tilde{\mathbb{P}}, \mathbb{P}^*) + d(\mathbb{P}, \mathbb{P}^*)$. We want to approximate the distance $d(\mathbb{P}, \tilde{\mathbb{P}})$ by the distance $d(\tilde{\mathbb{P}}, \mathbb{P}^*)$, which is the distance between Tree 1 and the process approximation Tree 2. Therefore we would like to guarantee, that $d(\mathbb{P}, \mathbb{P}^*)$ is small enough, i.e. $d(\mathbb{P}, \mathbb{P}^*) \leq \varepsilon$. Lemma 3.1 and Theorem 3.1 in [49] (originally in [38]) give us the bounds for the distance between stochastic process and a tree. It can be shown ([49]), that the upper bound for the nested distance is converging to zero when the bushiness of the tree is increasing. Bounds for the distance between stochastic process and a tree can be generalized for the case of two stochastic processes as the distance between two stochastic processes is the nested distance between two infinitely large trees (see [49], [37]).

6. Scenario and Scenario Tree Generation

In this chapter, the most common methods for generation of scenario trees will be presented.

6.1 Scenario Generation Methods

6.1.1 Conditional Sampling

Methods belonging to this group are one of the most common methods for scenario generation. At every node of the scenario tree, several values of the stochastic process $\{\boldsymbol{\omega}_t\}$ are sampled. We can sample directly from the distribution of or we can evolve the process according to some formula, e.g. $\boldsymbol{\omega}_{t+1} = z(\{\boldsymbol{\omega}_\tau, \tau < t\}, \varepsilon)$. As it is shown in [14] for *Markov structure of the data*, the vector autoregressive model of the first order can be used for conditional sampling of scenarios taking into consideration the already created structure of the scenario tree. We can write it in the form

$$\boldsymbol{\omega}_t = \mathbf{P}\boldsymbol{\omega}_{t-1} + \boldsymbol{\varepsilon}_t,$$

where $\mathbf{P}$ is the transition matrix, which can be dependent on t, and mostly $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma})$,. Here, $\boldsymbol{\omega}_t$ depends only on the preceding component $\boldsymbol{\omega}_{t-1}$ and on the additional random vector $\boldsymbol{\varepsilon}_t$, which is independent of the history $\boldsymbol{\omega}^{t-1,\bullet}$. In case of interstage independence, we have a special form of the Markov structure written above with $\mathbf{P}$ being a zero matrix. The Markov property enables us to sample directly form the probability distribution of $\boldsymbol{\varepsilon}_t$ at each node. Sometimes the distribution of $\boldsymbol{\varepsilon}_t$ is discretized at a given number of points and the obtained realizations are added to the already obtained past values of $\mathbf{P}\boldsymbol{\omega}_{t-1}$.

Another technique based on the Markov structure of the data is the *sequential importance sampling*, which takes into consideration the given suitably labeled tree structure already in the process of simulation. It is an iterative procedure based on a *scenario tree nodal partition matrix*, that uniquely describes the structure of the associated scenario tree. The rows of that matrix are equal to the number of scenarios and the number of columns are equal to the number of stages. For the iterative procedure, the matrix plays the role of the conditional scenario generator in an input and it is an output of the sampling algorithm. Initially the number of stages T, the maximum number of possible iterations, a stopping criterion and the initial scenario tree structure described by the corresponding nodal partition matrix must be specified. At every iteration, we specify and solve some version of the stochastic program given in the arborescent form (4.1.3), the nodal values of the importance sampling criterion are evaluated along the tree and a new structure is defined through and update of the nodal partition matrix. For more details about these procedure and the algorithm for sequential sampling, see [12] and references therein.

Sampling methods are not very suitable for multivariate random variables and the marginals are often sampled separately. Then the univariate components are combined all-against-all, which results in a vector of independent random variables. However, the size of the tree grows exponentially with the dimension of the random vector, for example sampling K scenarios and for m marginals, we obtain K^m scenarios.

Since the available data are often in the form of correlated multivariate time series of different lengths, with change points etc., we need to take into consideration the correlation and also to reduce the dimension, which leads to the reduction of the number of scenarios. One common approach is to find *the principal components* and sample those. Under normality assumption, *factor analysis* is mostly used. Both of these methods explain the correlation structure of the data by a small number of independent factors or components, which we can simulate separately. For more details see [30].

One possibility to improve the sampling method is to use integration quadratures or low discrepancy sequences as in [34]. The quadratures give a good approximations of given probability measures by a small number of quadrature points. In [27] the adjusted random sampling technique is considered. An even number of nodes is assumed and *antithetic sampling* is used to fit every odd moment of the underlying distribution. Antithetic sampling improves Monte Carlo technique by exploiting spatial structure, for more details see [33].

For sampling, the main source of instability or bias is the lack of scenarios. With an increasing number of scenarios, the discrete distribution converges to the true distribution. By increasing the number of scenarios, the trees will be closer to the true distribution and also closer to each other.

6.1.2 Sampling Based Methods Especially Suitable for Generating Multivariate Vectors

Some sampling-based methods can deal very well with generating multivariate correlated vectors using transformations. The marginal distributions, which can be arbitrary and even from different families, and the correlation matrix are specified by the user. They are usually estimated from historical data.

In [4] the main idea is transforming a multivariate normal random vector into the desired random vector. This method is especially suitable for the case when the marginal distributions of the component random variables are not from the same family of distributions. Suppose a random vector $\mathbf{X} = (X_1, \ldots, X_k)^T$ with arbitrary marginal distributions and a feasible correlation matrix. We take a random vector $\mathbf{Z}$ (base vector) with a known correlation matrix and we transform it in such way, that we achieve the desired marginal distributions for the components of the input vector $\mathbf{X}$. We adjust the correlation matrix of the random vector to obtain the target correlation matrix of $\mathbf{X}$. A standard multivariate normal random vector is used as the base vector in the paper, and $\mathbf{X}$ is referred as having a *NORTA* (NORmal To Anything) distribution.

Another approach on the similar basis in [7] deals with representing stationary multivariate time series with arbitrary marginal distributions and autocorrelation

structures. The idea here is transforming a Gaussian vector autoregressive process into the desired multivariate time-series input process. The correlation structure of the Gaussian vector autoregressive process adjusted to obtain the desired correlation structure for the simulation input process.

Another approach is using *copulas* for generation of scenarios with multivariate structure. According to Sklar [46], copula is "a function that links a multidimensional distribution to its one-dimensional margins". As in [26], an n-dimensional copula is the joint cumulative distribution function (cdf) of any n-dimensional random vector with standard uniform marginal distributions, function $C : [0,1]^n \to [0,1]$. According to *Sklar's theorem*, for any n-dimensional cdf F with marginal distribution functions $F_1, \ldots, F_n$, there exists a copula C, such that

$$F(x_1, \ldots, x_n) = C(F_1(x_1), \ldots, F_n(x_n)).$$

Furthermore, for continuous marginal cdfs F_i, C is unique (proof in [47]). As a consequence, we have that for every $\mathbf{u} = (u_1, \ldots, u_n) \in [0,1]^n$, it holds

$$C(u_1, \ldots, u_n) = F(F_1^{-1}(u_1), \ldots, F_n^{-1}(u_n)),$$

where F_i^{-1} is the generalized inverse of F_i. Copula does not change under strictly increasing transformations of the margins, therefore we can transform margins form one continuous distribution to another without changing the copula. Also any statistical property dependent only on the copula is invariant to strictly increasing transformations of the margins, e.g. the Spearman's rank correlation. We call a discrete distribution described by a matrix of equiprobable outcomes $\mathbf{X} = (x_{is})$ an *empirical distribution*. Its marginal cdfs are given by the formula

$$F_i^e(x) = \frac{|\{s : x_{is} \leq x\}|}{n_S},$$

where n_S is the number of scenarios (samples) and $|\cdot|$ denotes the cardinality of a set. If we assume x_{is} to be distinct in every margin $\mathbf{x}_i$, the cdfs evaluated at the sample points x_{is} are equal to

$$F_i^e(x_{is}) = \frac{rank(x_{is}, \mathbf{x}_i)}{n_S},$$

where $rank(x_s, \mathbf{x})$ is the order of value x_s in a vector $\mathbf{x}$ with values between 1 and n_S. For the copula of the empirical distribution, called *empirical copula*, evaluated at the sample points, we have

$$\begin{aligned} C\left(\tfrac{k_1}{n_S}, \ldots, \tfrac{k_n}{n_S}\right) &= \tfrac{1}{n_S}\left|\left\{s : x_{is} \leq x_{(ki)} \quad \text{for} \quad \forall\, i \in \{1, \ldots, m\}\right\}\right| \\ &= \tfrac{1}{n_S}\left|\{s : rank(x_{is}) \leq k_i \quad \text{for} \quad \forall\, i \in \{1, \ldots, n\}\}\right|, \end{aligned}$$

where x_k is the k^{th}-smallest element of vector $\mathbf{x}$. The empirical copula is uniquely described in terms of ranks of the original sample.

In [26] a particular scenario generation method, as it is presented below, is described. Our goal is to generate n_S samples from a given n-variate distribution, e.g. a matrix $\mathbf{X} \in \mathbb{R}^{n,n_S}$ of outcomes. The empirical copula associated with the outcomes is called *a scenario copula*. The following method consists of two parts.

1. *Create a scenario copula*, described in terms of the ranks of the margins. We want to have a set of n_S scenarios, each consisting of the ranks of values we want to use from each of the n margins., e.g. "take the maximum of margin 1, take the second-smallest value of margin 2, etc.". There are several ways to do that

   - *Sampling* from the true distribution or its approximation and computing the ranks of the values, one margin at a time. For each margin, the values of the outcomes are replaced by their ranks inside the vector of all the outcomes for the given margin.
   - *Parametric family of copulas* with parameters estimated from historical data are used.
   - *Optimization approach to directly couple the ranks* in a way that minimizes some distance from the target distribution.

2. *Generate the values of each margin.*

   - *Use a prescribed discretization of the marginal distributions,* e.g. if the marginal cdfs F_i are known, we start with some discretization $\{u_1, \ldots, u_{n_S}\}$ of the standard uniform distribution and we let $x_{is} = F_i^{-1}(u_s)$. As u_s it is commonly used $u_s = \frac{s}{n_S+1}$ or $u_s = \frac{2s-1}{2n_S}$.
   - *Compute marginal moments from the historical data* and use a transformation based moment-matching method to transform the scenarios to match the moments. For the first four moments a cubic transformation can be used, e.g. as in [23].

Having the structure (copula) and the values of margins, we connect the margins in the way specified by the coupling of ranks.

We obtain the copula from the joint cdf by transforming the margins to the standard uniform distribution, so it can be viewed as the joint distribution stripped of all information about the margins. None of the information about the multivariate structure is lost by transforming the margins. Therefore we can decouple the margins from the multivariate structure and these two can be modeled independently. In [26] a particular scenario-generation method using copulas is also presented. In general, the goal is to generate n_S samples from a given n-variate distribution, which means a matrix $\mathbf{X} \in \mathbb{R}^{n,n_S}$ of outcomes. A particular method for scenario-generation using copulas is presented in [26].

Using copulas to construct joint distributions and pairwise correlation to incorporate dependence among the variables is also used in [5]. That approach is especially designed to permit the use of expert's subjective judgements of marginal distributions and correlations.

6.1.3 Moment-matching Methods

If the distribution functions of the marginals are not known, the marginals can be described by their moments, most often by the first four moments (mean,

variance, skewness, curtosis). The correlation matrix is specified by the user (usually estimated from historical data) and sometimes other statistical properties such as percentiles can be specified. The moment-matching methods construct a discrete distribution with these specified properties.

In [48] a construction of an n-point discrete distribution that matches the first $2n-1$ moments of any continuous distribution is shown. For constructing the moment-matching discrete distributions, one has to choose the points of the distribution to be the roots of an orthogonal polynomial. Then, the orthogonal polynomials define a moment-matching discrete distribution.

For generating a limited number of discrete outcomes that satisfy specified properties, a method based on minimizing some measure of distance between the statistical properties of the generated outcomes and the specified properties is used in [24]. The set of all specified statistical properties is denoted by S and the specified value of statistical property i in S is denoted by S_{VAL_i}. We denote by I the number of random variables, by T the number of stages and by N_t the number of conditional outcomes in stage t. For simplicity, a symmetrical tree is assumed (the number of branches is the same for all conditional distributions at the same period). We define $\mathbf{x}$ to be the outcome vector of dimension $I \times N_1 + I \times N_1 \times N_2 + \ldots$, $I \times N_1 \times N_2 \times \ldots \times N_T$, and p a probability vector of dimension $N_1 + N_1 \times N_2 + \ldots + N_1 \times N_2 \ldots N_T$. The mathematical expression for statistical property i in S is denoted by $h_i(\mathbf{x}, \mathbf{p})$. We define a matrix $\mathbf{M}$ of zeros and ones, whose number of rows is equal to the length of $\mathbf{p}$ and whose number of columns is equal to the number of nodes in the scenario tree, where each column is the indicator of a conditional distribution at one node. Each column of $\mathbf{M}$ extracts a conditional distribution in the scenario tree. We set weights w_i for statistical property i in S. Our aim is to construct $\mathbf{x}$ and $\mathbf{p}$ so that the statistical properties of the approximating distribution match as well as possible the specified statistical properties. This is done by minimizing a measure of distance between the statistical properties of the constructed distribution and the specifications, subject to constraints for probabilities (nonnegativity and summing up to one). To measure distance, we use the square norm, i.e. $\min_{\mathbf{x},\mathbf{p}} \sum_{i \in S} (h_i(\mathbf{x}, \mathbf{p}) - S_{VAL_i})^2$.

This idea is adopted in [18] and further developed and used to generate the scenario tree, which is input to the financial portfolio allocation problem. The ideas of *sequential optimization* (scenario tree is constructed by considering the branching at each node separately) and *overall optimization* (we consider all nodes of the scenario tree and generate the whole tree) are presented there.

A heuristic for moment-matching scenario generation is introduced in [23], especially and algorithm to obtain discrete joint distribution having specified first four marginal moments and correlations. The multivariate problem is decomposed into univariate ones and then we use an iterative procedure (simulations, Cholesky decomposition of the correlation matrix, cubic and matrix transformations) to get the correct correlations, but keep the marginal moments at the same time.

The paper [31] focuses on generating multivariate data that have moments arbitrary close to the desired ones. They can be generated as linear combinations of variables with known theoretical moments.

The smoothness of the distribution usually increases with the number of scenarios, but however, this is not true for all moment-matching methods, e.g. it works for transformation based methods, but doesn't work for optimization based me-

thods. Generally, the first four moments are good enough for description of the marginals. For the multivariate structure, the correlation structure is not enough, therefore sometimes higher co-moments are tried to be matched or copulas are used.

6.1.4 Path-based methods

If the scenarios are generated by procedures that do not consider the tree structure of the input data, we need another steps to build a scenario tree of the prescribed structure. Sometimes, the generated set of data paths is used to build a scenario tree by cutting and pasting the data in an intuitive way. Another way is applying *cluster analysis*. The clustering can be done according to the first component (or subvector) ω_1 of the stochastic process ω and then it continues by conditional clustering according to the second component (or subvectors) ω_2 of the objects that are included into the previously created clusters, etc. If we take into considerations the interstage dependences as in [12], [14], we use instead a *multi-level clustering scheme*, which uses the whole sequences of the data $(\omega_1, \ldots, \omega_T)$. The procedure can be written as follows:

- First the dissimilarity measure for each pair ω^i, ω^j is calculated,

$$d(\omega^i, \omega^j) = \sum_{t=1}^{T} w_t \|\omega_t^i - \omega_t^j\|,$$

  where $w_t \geq 0$ are nonincreasing weights. Usually more emphasis is given on differences at the beginning of the sequence.

- We use the measures of dissimilarity among the compared objects in definitions of the standard measures of dissimilarity of clusters and then we use it subsequently in the cluster analysis approach. This results in K_1 clusters, $C_1^1, \ldots, C_1^{K_1}$, which are represented by $\tilde{\omega}_1^k, k = 1, \ldots, K_1$, that can be the mean values of the first components ω_1 of the observations ω, that are included in the cluster. The probabilities of $\tilde{\omega}_1^k, k = 1, \ldots, K_1$, are equal to the sum of probabilities of the individual ω^i's belonging to the relevant cluster.

- The clustering procedure continues for each cluster C_1^k separately, it starts with the second component ω_2 of the observations included into C_1^k, or equivalently with the first component ω_1 replaced by $\tilde{\omega}_1^k$ and so on.

Another approach is *randomized clustering* in [18]. The randomized clustering algorithm is repeated until acceptable clustering is found. The algorithm can be described in the following steps:

- *Step 1 - Initialization.* Create a root node, with N scenarios. Initialize all the scenarios (including the centroid) with the desired starting point (e.g. today's prices). We form a job queue consisting of the root node.

- *Step 2 - Simulation.* Remove one node from the job queue and simulate one time period of growth (e.g. from today's to tomorrow's prices) in each scenario.

- *Step 3 - Randomized seeds.* We choose randomly a number of distinct scenarios around which to cluster the rest, one per desired branch in the scenario tree.

- *Step 4- Clustering.* Group each scenario with the seed point to which it is the closest. If the resulting clustering is unacceptable, go back to *Step 3.*

- *Step 5 - Centroid selection.* For each cluster, find the scenario which is the closest to its center and take it as centroid.

- *Step 6 - Queueing.* Create a "child"scenario tree node for each cluster (whose probability is proportional to the number of scenarios in the cluster), and install its scenarios and centroid. In case the child nodes are not leaves, we append to the job queue. For a non-empty queue, return to *Step 2.* Otherwise *END.*

If we want the current node to have k following branches, then we need to form k clusters. The seed points around which we build clusters might be chosen to be the first k scenarios, because scenarios are independently generated and they are in an arbitrary order. For determining which scenario is the closest to a particular seed, we can use as a distance measure for instance Euclidean, Manhattan or some p-norm. Each cluster is represented with a single point, which becomes the data in the scenario tree, usually it is mean, median, center of gravity etc. The seed point may be an outlying event that gathered a cluster of all the points that were away from the center in similar direction etc., therefore it is not very suitable to take it as a "center".

6.1.5 Methods Based on Distance

For multistage models approximated by trees, we will show *the deterministic iteration* and *the stochastic approximation*, as they are introduced in [35]. We will consider two probability distribution functions $P(\mathbf{u})$ and $\hat{P}(\mathbf{u})$ on $\mathbb{R}^T$, $\mathbf{u} = (u_1, \ldots, u_T)$. We consider the distance

$$\|\mathbf{u} - \mathbf{v}\| = \sum_{t=1}^{T} a_t |u_t - v_t|,$$

where $(a_1, \ldots, a_T)$ is a vector of nonnegative weights, which show the importance of the particular dimensions, very often it is $a_1 \geq a_2 \geq \ldots \geq a_T > 0$. We introduce

$$L_p = \inf\{L : |f(\mathbf{u} - f(\mathbf{v})| \leq L \sum_{t=1}^{T} a_t |u_t - v_t| \max(1, |u_t|^{p-1}, |v_t|^{p-1})\}$$

and

$$d_p(P,\hat{P}) = \sup\left\{\int f(\mathbf{u})\,dP(\mathbf{u}) \;-\; \int f(\mathbf{u})\,d\hat{P}(\mathbf{u}) \;:\; L_p(f) \le 1\right\}.$$

The aim is to approximate the distribution P by the discrete multivariate distribution $\hat{P}$.

Assume a tree of with height T. At level $0, 1, 2, 3, \ldots, T$ there are $1, k_1, k_2, k_3, \ldots, k_T$ nodes respectively. The nodes are denoted by $\{(t,i) \;:\; 1 \le i \le k_t; 1 \le t \le T\}$ (for the root of the tree we have $(0,0)$). For $0 \le t < s \le T$ and node (t,j) being a predecessor of the node (s,i), we denote that fact here by $pred_t(s,i) = j$. With each node of the tree, a real value $z_{t,i}$ is associated. We denote the vector of these real values by $Z = \{z_{t,i} \;:\; i = 1, \ldots, k_t; t = 1, \ldots, T\}$. The points in Z can be grouped into vectors $\mathbf{z}_1, \ldots, \mathbf{z}_{k_T}$ as $\mathbf{z}_i = \left(z_{1,pred_1(T,i)}, z_{2,pred_2(T,i)}, \ldots, z_{T-1,pred_{T-1}(T,i)}, z_{T,i}\right)$. We denote by $(p_i \;:\; i = 1, \ldots, k_T)$ the discrete probability measure sitting on vectors $\mathbf{z}_i$. Assigning probabilities to the terminal nodes of the tree is sufficient, because the probabilities of the nonterminal nodes $(p_{t,j})$ can be calculated as $p_{t,j} = \sum_{\{i:j=pred_t(T,i)\}} p_i$.

The tree represents the discrete stochastic process $(\omega_1, \ldots, \omega_T)$, the probability of $(\omega_1, \ldots, \omega_T) = \mathbf{z}_i$ is equal to p_i and for marginals, the probability of $\omega_t = z_{t,j}$ is equal to $p_{t,j}$. The mass points Z induce a partition $\mathcal{E}^{(Z)} = \{E_i^{(Z)}; i = 1, \ldots, k_T\}$ on $\mathbb{R}^T$:

$$E_i^{(Z)} = \{\mathbf{u} \in \mathbb{R}^T \;:\; \|\mathbf{u} - \mathbf{z}_i\| = \min_j \|\mathbf{u} - \mathbf{z}_j\|\}.$$

The partitions are not disjoint, but we can make them disjoint. If $\mathbf{u}$ is in several sets $E_i^{(Z)}$, we assign it to the set with minimal i. We denote the distribution sitting on points $\mathbf{z}_i$ with masses $p^{(Z)}$, where

$$p_i^{(Z)} = \int\limits_{E_i^{(Z)}} dP(\mathbf{u}),$$

by $\tilde{P}^{(Z)}$. We will consider

$$D(Z) = \int \min_j \|\mathbf{u} - \mathbf{z}_j\| dP(\mathbf{u}) = \sum_{i=1}^{k_T} \int\limits_{E_i^{(Z)}} \|\mathbf{u} - \mathbf{z}_i\| dP(\mathbf{u}).$$

The aim is to minimize $D(Z)$ over all tree structures Z.

The Deterministic Iteration Algorithm

For P completely known, the following algorithm as presented in [35] can be used.

- *Step 1.* Initialize $s = 0$ and $Z^{(0)} = \left\{z_{t,i}^{(0)} \;:\; 1 \le i \le k_t; 1 \le t \le T\right\}$.
- *Step 2.* Find $E_i^{(Z^{(s)})}$ for $i \le t \le k_T$.

- *Step 3.* Let $z_{t,i}^{(s+1)} \in \arg\min_y \left\{ \sum_{\{j:pred_t(j)=i\}} \int_{E_j^{(Z^{(s)})}} |u_t - y| dP(\mathbf{u}) \right\}$, which means that $z_{t,i}$ is the median of the distribution P restricted to

$$\bigcup_{\{j:pred_t(j)=i\}} E_j^{(Z^{(s)})}$$

in the t^{th}-coordinate.

- *Step 4.* If $D(Z^{(s)}) \geq D(Z^{(s-1)})$, then *STOP*. Otherwise put $s := s+1$ and *GOTO Step 2.*

It could be shown (see Proposition, p. 13 in [35]), that if $Z^{(s)}$ is a sequence of trees generated by the deterministic iteration algorithm, then it holds $D(Z^{(s+1)}) \leq D(Z^{(s)})$. If we have equality, $D(Z^{(s^*+1)}) = D(Z^{(s^*)})$, for some s^*, then $D(Z^{(s+1)}) = D(Z^{(s)})$ for $s \geq s^*$ and $\nabla_Z D(Z^{(s^*)}) = 0$.

The Stochastic Approximation Algorithm

This algorithms aims at estimating the best approximation in terms of distance on the basis of sample from P. We consider an i.i.d. sequence of vectors $\boldsymbol{\omega}^{(s)} = (\omega_1^{(s)}, \ldots, \omega_T^{(s)}), s = 1, 2, \ldots$, each of them has a probability distribution P. Then the *unconstrained stochastic approximation algorithm* according to [35] is

- *Step 1.* Initialize $s = 0, \quad Z^{(0)} = \left\{ z_{t,i}^{(0)} : 1 \leq i \leq k_t; 1 \leq t \leq T \right\}$ and $p_i^{(0)} = 1/k_T$ for $1 \leq i \leq k_T$.

- *Step 2.* Observe the next trajectory $\omega^{(s)}$.

- *Step 3.* Find $j \in \{1, 2, \ldots, k_T\}$ such that $\omega^{(s)} \in E_j^{(Z^{(s)})}$.

- *Step 4.* For $1 \leq t \leq T$ change

$$z_{t,pred_t(j)}^{(s+1)} = z_{t,pred_t(j)}^{(s)} + \begin{cases} \frac{a_t}{s} & \text{if} \, \omega_t^{(s)} \geq z_{t,pred_t(j)}^{(s)} \\ -\frac{a_t}{s} * & \text{if} \, \omega_t^{(s)} < z_{t,pred_t(j)}^{(s)}, \end{cases}$$

Other points are left unchanged.

- *Step 5.* Estimate the probabilities

$$p_j^{(s+1)} = \frac{s p_j^{(s)} + 1}{s+1}$$

$$p_i(s+1) = \frac{s p_i^{(s)}}{s+1} \qquad \text{for} \, i \neq j$$

- *Step 6.* Put $s := s+1$ and *GOTO Step 2.*

It is shown, that if $Z^{(s)}$ is a sequence of trees generated by the stochastic approximation algorithm, then $\nabla_Z D(Z^{(s)}) \to 0\, a.s.$, and if $D(Z)$ has a unique minimizer Z^*, then $Z(s) \to Z^*\, a.s.$

Additionally, some scenario trees have to fulfill some consistency relations, e.g. for modeling interest rates, the scenario process should be consistent with today's observations of forward rates. These relations constrain the set of allowed trees, therefore they make the approximation problem a constrained problem (for the *constrained stochastic approximation algorithm* see [35]). In [49] a method for distribution quantization suitable for multistage stochastic optimization programs taking into account both the stochastic process and the stagewise process is described.

6.2 Scenario Reduction and Related Methods

Since the size of the scenario tree can be very large, there are several methods of decreasing it. In the next subsection, the methods try to find a scenario subset of prescribed cardinality, and a probability measure based on this set, which is the closest to the initial distribution in terms of probability metrics.

6.2.1 Scenario Reduction Using Metrics

We suppose a probability distribution P on $\mathbb{R}^n$ of a n-dimensional stochastic data process $\omega = \{\omega_t\}_{t=1}^T$ with a finite support consisting of K scenarios $\omega^i = \{\omega_t^i\}_{t=1}^T, i = 1, \ldots, K$, and their probabilities $p_i, \sum_{i=1}^K p_i = 1$. The idea of scenario reduction is to determine a probability distribution Q, which will best approximate the probability distribution P in terms of a certain probability distance between P and Q, but whose support will consist of a subset of $\{\omega^1, \ldots, \omega^K\}$. As a distance, the Kantorovich distance is used. We will use the concept of [17]. If we assume Q being a distribution of another n-dimensional stochastic process $\bar{\omega}$ with scenarios $\bar{\omega}^j$ and probabilities $q_j, j = 1, \ldots, \bar{K}$. Then we will write the distance as

$$d_K(P,Q) = \inf\left\{\sum_{i=1}^{K}\sum_{j=1}^{\bar{K}} \eta_{i,j} c_T(\omega^i, \bar{\omega}^j) : \eta_{i,j} \geq 0, \sum_{i=1}^{K} \eta_{i,j} = q_j, \sum_{j=1}^{\bar{K}} \eta_{i,j} = p_i, \text{for } \forall i, j\right\},$$

where we have $c_t(\omega^i, \bar{\omega}^j) := \sum_{\tau=1}^t |\omega_\tau^i - \bar{\omega}_\tau^j|, t = 1, \ldots, T$ and $|\cdot|$ is some norm on $\mathbb{R}^n$, hence we have that c_T measures the distance between scenarios on the whole time horizon. The support of the reduced probability distribution Q of ω consists of scenarios ω^j for $j \in \{1, \ldots, K\} \backslash J$, where J is some index set of deleted scenarios. For fixed $J \subset 1, \ldots, K$, the minimal distance is

$$d_K(P,Q) = \sum_{j \in J} p_i \min_{j \notin J} c_T(\omega^i, \omega^j)$$

and the probability q_j of the preserved scenarios $\omega^j, j \notin J$, of Q is given by the *optimal redistribution rule*

$$q_j := p_j + \sum_{i \in J(j)} p_i,$$

where we have $J(j) := \{i \in J : j = j(i)\}$, $j(i) \in \arg\min_{j \notin J} c_T(\omega^i, \omega^j)$ for $\forall i \in J$. The new probability of the preserved scenario is equal to the sum of its former probability and of all probabilities of deleted scenarios, which are the closest to this preserved one in terms of c_T. (Deleted scenarios have probability zero.) For fixed cardinality $card(J)$, the optimal choice of an index set J for scenario reduction can be computed from the following *optimal reduction problem*

$$\min \left\{ \sum_{i \in J} p_i \min_{j \in J} c_T(\omega^i, \omega^j) : J \subset 1, \ldots, K, card(J) = K - s \right\},$$

where $s = K - card(J)$ is the number of preserved scenarios.

We want to determine a reduced probability distribution Q of ω, such that the set of deleted scenarios has maximal cardinality and Q is close to the original distribution P with a given accuracy $\varepsilon > 0$, i.e. $d_K(P, Q) < \varepsilon$. We can formulate *the maximal reduction strategy* as determining an index set J with maximal cardinality $card(J)$, such that $\sum_{i \in J} p_i \min_{j \notin J} c_T(\omega^i, \omega^j) \leq \varepsilon$. The probabilities $q_j, j \notin J$, of the preserved scenarios are given by the redistribution rule.

For solving the optimal reduction problem, heuristic algorithms were developed. For the cases when $card(J) = 1$ and $card(J) = K - s$, the optimal reduction problem can be solved as follows:

- Deleting one scenario. For $card(J)$, the optimal reduction problem has the form $\min_{l \in \{1,\ldots,K\}} p_l \min_{j \neq l} c_T(\omega^l, \omega^j)$. Suppose, that the minimum is attained at $l_* \in \{1, \ldots, K\}$. We delete scenario ω^{l_*}, and from the redistribution rule we have the reduced probability distribution Q. For $j_* \in \arg\min_{j \not= l_*} c_T(\omega^{l_*}, \omega^j)$, $q_{j_*} = p_{j_*} + p_{l_*}$ and $q_l = p_l$ for $\forall l \notin \{l_*, j_*\}$.

- Optimal selection of a single scenario. For $card(J) = K - 1$, the optimal reduction problem is of the form $\min_{u \in \{1,\ldots,K\}} \sum_{i=1}^{K} p_i c_T(\omega^i, \omega u)$. For minimum attained at $u_* \in \{1, \ldots, K\}$, we keep only the scenario ω^{u_*} and from the redistribution rule we have $q_{u_*} = p_{u_*} + \sum_{i \neq u_*} p_i = 1$.

The optimal deletion of a single scenario can be repeated recursively until the prescribed number $K - s$ of scenarios is deleted (*backward reduction*). On contrary, if the number of preserved scenarios is relatively small, the optimal selection of a single scenario can be repeated recursively until we reach the prescribed number s of preserved scenarios (*forward selection*).

Simultaneous Backward Reduction

A particular variant of the backward algorithm as described in [17] is

- *Step 0.* Computation of distances of scenario pairs $c_{kj} := c_T(\omega^k, \omega^j)$, $k, j = 1, \ldots, K$.
  Sort the computed distances $\{c_{kj} : j = 1, \ldots, K\}$, for each $k = 1, \ldots, K$.

- *Step 1.* Compute $c_{ll}^{[1]} := \min\limits_{j \neq l} c_{lj}$, $l = 1, \ldots, K$ and $z_l^{[1]} := p_l c_{ll}^{[1]}$, $l = 1, \ldots, K$.
  Choose $l_1 \in \arg\min\limits_{l \in \{1, \ldots, K\}} z_l^{[1]}$.
  Set $J^{[1]} := \{l_1\}$.

- *Step i.* Compute $c_{kl}^{[i]} := \min\limits_{j \notin J^{[i-1]} \cup \{l\}} c_{kj}$ for $l \notin J^{[i-1]}, k \in J[i-1] \cup \{l\}$ and
  $z_l^{[i]} := \sum\limits_{k \in J^{[i-1]} \cup \{l\}} p_k c_{kl}^{[i]}, l \notin J^{[i-1]}$.
  Choose $l_i \in \arg\min\limits_{l \notin J^{[i-1]}} z_l^{[i]}$.
  Set $J^{[i]} := J^{[i-1]} \cup \{l_i\}$.
  Do for $i = 2, \ldots, K - s$.

- *Step(K-s+1).* $J := J^{[K-s]}$ is the index set of the deleted scenarios. The optimal probabilities for the preserved scenarios are computed from the optimal redistribution rule.

Fast Forward Selection

This algorithm, described in [17], works as follows

- *Step 0.* Computation of distances of scenario pairs $c_{ku}^{[1]} := c_T(\omega^k, \omega^u)$, $k, u = 1, \ldots, K$.

- *Step 1.* Compute $z_u^{[1]} := \sum\limits_{k=1, k \neq u} p_k c_{ku}^{[1]}$, $u = 1, \ldots, K$.
  Choose $u_1 \in \arg\min\limits_{u \in \{1, \ldots, K\}} z_u^{[1]}$.
  Set $J^{[1]} := \{1, \ldots, K\} \backslash \{u_1\}$.

- *Step i.* Compute $c_{ku}^{[i]} := \min\left\{c_{ku}^{[i-1]}, c_{ku_{i-1}}^{[i-1]}\right\}$, $k, l \in J^{[i-1]}$ and
  $z_u^{[i]} := \sum\limits_{k \in J^{[i-1]} \backslash \{u\}} p_k c_{ku}^{[i]}$, $u \in J^{[i-1]}$.
  Choose $u_i \in \arg\min\limits_{u \in J^{[i-1]}} z_u^{[i]}$.
  Set $J^{[i]} := J^{[i-1]} \backslash \{u_i\}$. Do for $i = 2, \ldots, s$.

- *Step (s+1).* $J := J^{[K-s]}$ is the index set of deleted scenarios. The optimal probabilities for the preserved scenarios are computed from the optimal redistribution rule.

We assume a finitely many paths or scenarios $\omega^i = \{\omega_t^i\}_{t=1}^T$ and the corresponding given probabilities $p_i, i = 1, \ldots, K$ of an n-dimensional stochastic process $\omega = \{\omega_t\}_{t=1}^T$. We assume that at $t = 1$, $\omega_1^1 = \ldots = \omega_1^K =: \omega_1^*$. Therefore, we will regard $t = 1$ as the root node of a scenario tree with K branches (ω^i form a fan

of scenarios). The following scenario tree construction is based on implementing a backward strategy using the scenario reduction principle. On the time horizon $\{1,\dots t\}$ at each $t=1,\dots T$ as a similarity measure. The number of nodes of the fan of individual scenarios is recursively reduced by modifying the tree structure and by bundling scenarios. The Kantorovich distance of the original and the reduced (sub)trees on $\{1,\dots,t\}, t=T,T-1,\dots,2,1$ is compared and scenarios are deleted, if the reduced tree is still close to the original one. The scenario sets deleted at t are denoted by J_t and the set of scenarios preserved at t is denoted by I_t. the following algorithm uses these ideas.

Scenario Tree Construction Using Scenario Reduction Principle

Suppose the tolerances $\varepsilon_t > 0, t=1,\dots,T$ are given.

- *Step 1.* Apply the maximal reduction strategy and the simultaneous backward reduction to determine the index set $J_T \subset \{1,\dots,K\} = I_{T+1}$, such that
$$\sum_{i\in J_T} p_i \min_{j\notin J_T} c_T(\omega^i,\omega^j) \le \varepsilon_T.$$
Set $I_T := I_{T+1}\backslash J_T$ and $\omega^i_{app} := \omega^i, i\in I_T$. From the optimal redistribution rule calculate the optimal probabilities $\pi^i_T, i\in I_T$ for the preserved scenarios.

- *Step (T-t+1).* Do this step for $t=1,\dots T-1$.
Reduction.
Apply the maximal reduction strategy and the simultaneous backward reduction to determine the index set $J_t \subset I_{t+1}$, such that
$$\sum_{i\in J_T} p_i \min_{j\in I_{t+1}\backslash J_t} c_t(\omega^i,\omega^j) \le \varepsilon_t.$$
Set $I_t := I_{t+1}\backslash J_t$.
Scenario bundling.
For each $j\in J_t$ select an index $i^*\in \arg\min_{i\in I_t} c_t(\omega^i,\omega^j)$, add π^j_{t+1} to $\pi^{i^*}_{t+1}$ and bundle scenario j with i^*:
$\omega^j_{t,app} := \omega^{i^*}_t \quad \tau=2,\dots,t,$
$\omega^j_{t,app} := \omega^j_t \quad \tau=t+1,\dots,T.$
Set $\omega^i_{t,app} := \omega^i_{t+1,app}$, $\pi^i_t := \pi^i_{t+1}, i\in I_t$.

- *Step k = T.* Set $\omega^i_{1,app} := \omega^*_1$ and consider the tree consisting of scenarios $\{\omega^i_{t,app}\}_{t=1}^T$ for $i\in I_T$

.

Another papers focusing on scenario reduction and generation of scenario trees using the idea of distances are [41], [13], [21] and [20].

6.2.2 Related Methods

There are many other methods for treating scenario trees. In [54], it is supposed that for every scenario $k, k = 1, \ldots, K$, there is an optimal solution $\mathbf{x}^k$ of the multistage stochastic program, we call it *scenario-solution*. When all scenario-solutions are computed, we analyze them for discovering general trends, clusters of solutions, etc.. The purpose is to obtain one solution. Therefore, we assign nonnegative weights summing up to 1 to the particular scenario-solutions, $w^k, k = 1, \ldots, K$. We arrive at solution $\hat{\mathbf{x}} = \sum_{k=1}^{K} w^k \mathbf{x}^k$, called *average solution*. The weights can rely on expert's opinion (e.g. assigning importance to the various scenarios) or they can be probabilities of realizations of particular scenarios. A solutions that hedges against all eventualities, could be obtained by solving a problem of the type

$$\begin{aligned} & \min \sum_{k=1}^{K} w^k f(\mathbf{x}, \boldsymbol{\omega}^k) \\ \textit{subject to} \quad & \mathbf{x} \in \bigcap_{k=1}^{K} C^k, \end{aligned} \tag{6.2.2.1}$$

where C^k is the set of feasible solutions determined by the scenario-dependent constraints. The optimal solution of the above stated problem (6.2.2.1) is denoted by $\mathbf{x}^*$.

We can refer to the average solution $\hat{\mathbf{x}}$ as *implementable* solution. The implementable solution is not necessarily feasible. A solution, which is feasible for each particular scenario problem ($\mathbf{x} \in \bigcap_{k=1}^{K} C^k$), is *admissible*. In [54] a method of aggragating the scenario-solutions in an overall solution that converges to the solution of the problem (6.2.2.1) is presented.

Another method, that works iteratively, is in [8]. It solves the problem with a current scenario tree, add or remove some scenarios and solve the problem again. The procedure decides, where to add or remove a scenario based on the "importance of scenarios"measured by the expected value of perfect information.

If we want to reduce the number of nodes of the scenario tree, we can *recombine* the some nodes as in [28]. Two nodes of the scenario tree can be recombined at time t, if both trees share the same associated subtree.

7. Applications of Multistage Stochastic Programs

Multistage stochastic programs have large variety of applications in real-life situations, mainly in the field of finance and industry. In this chapter, some of the main applications will be mentioned.

7.1 Application of Multistage Stochastic Programs in Economics and Finance

Portfolio Optimization

One very common and natural application of multistage stochastic programming theory is portfolio optimization. It is assumed that the uncertainty is described by a discrete probability distribution of random parameters carried by finite number of scenarios with prescribed probabilities. We also suppose that the discrete probability distribution is an acceptable substitute of the true underlying probability distribution. For scenario ω^k we will denote the related coefficient and decision variables by the superscript k. We will present the structure of portfolio optimization models as in [10]. The main investment decision consists of selection of asset categories and wealth allocation over time. The outcome will be evaluated at time T_0, which is an endpoint of an interval $[0, T_o]$, which is discretized, i.e. it is covered by nonoverlapping time intervals indexed by $t = 1, \ldots, \tau$. The investor first constructs a portfolio at time 0 (at the beginning of the first period), and at the beginning of sebsequent periods $t = 2, \ldots, \tau$, he rebalances it in order to cover the target ratio or to contribute to the maximization of the final performance at T_0. In the general setting of a T-stage stochastic program, $\tau = T$, after T_0 we do not allow any further decisions. Stages do not necessarily correspond to time periods. The most demanding is the first-stage decision, which consists of all decision that have to be selected before any further information is known, it is just based on the already known probability distribution P, i.e. based on the already designed scenario tree. The second stage decisions are allowed to adapt to an additional information, which is revealed at the end of the first-stage period. To make it simpler, we will not distinguish strictly between indices of stages and time periods.

The decision variable $h_i^k(t)$ represents the holding in asset category i, at the beginning of time period t under scenario k, *after* the rebalancing decisions were made (the initial holding is $h_i(0)$). In the model, it can be the amount of money invested in i at the beginning of time period t (in dollars of the initial purchase price, in number of securities etc.). The value of the holdings at the end of period t might be affected by market returns ρ, we consider the expected return $r_i^k(t)$ of asset in category i under scenario k for period t. The wealth accumulated for the asset category i at the *end* of the t^{th}-period *before* the next rebalancing is done

is

$$w_i^k(t) := (1 + r_i^k(t))h_i^k(t), \quad \text{for}\, \forall i, t, k.$$

The purchases of assets are denoted by $b_i^k(t)$ and the sales as $s_i^k(t)$. We assume time-independent transaction costs α_i and symmetry in the transaction costs (purchasing one unit of asset category i at the beginning of period t requires $1 + \alpha_i$ units of cash, selling one unit of asset category i results of $1 - \alpha_i$ units of cash). For each asset category (except for cash), scenario and time period, we can write the *flow balance constraint* as

$$h_i^k(t) = (1 + r_i^k(t-1))h_i^k(t-1) + b_i^k(t) - s_i^k(t).$$

For cash, $i = 0$, the *flow balance equation* for each time periods and all scenarios is

$$\begin{aligned} h_0^k(t) = h_0^k(t-1)(1 + r_0^k(t-1)) + c^k(t) + \sum_{i=1}^{I} s_i^k(t)(1 - \alpha_i) - \\ - \sum_{i=1}^{I} b_i^k(t)(1 + \alpha_i) + \sum_{i=1}^{I} f_i^k(t)h_i^k(t) - y^{k-}(t-1)(1 + \delta^k(t-1)) - \\ - L^k(t) + y^{k-}(t), \end{aligned}$$

where $f_i^k(t)$ is the cash flow generated by holding one unit of the asset i during period t (coupons, dividends, etc.) under scenario k and $L^k(t)$ is the paydown of the committed liabilities in period t under scenario k. The borrowing in period t under scenario k at the borrowing rate $\delta^k(t)$ is denoted by $y^{k-}(t)$. The decision variables concerning the structure of external cash flows in period t under scenario k are $c^k(t) = c^{k+}(t) - c^{k-}(t)$. It is assumed, for simplicity, that all borrowing is done on a single period basis.

For holdings, purchases and sales expressed in numbers or in face values, the cash balance equation contains purchasing $\xi_i^k(t)$ and selling prices $\zeta_i^k(t)$, $\xi_i^k(t) > \zeta_i^k$,:

$$\begin{aligned} h_0^k(t) = h_0^k(t-1)(1 + r_0^k(t-1)) + c^k(t) + \sum_{i=1}^{I} \zeta_i^k(t)s_i^k(t) - \\ - \sum_{i=1}^{I} \xi_i^k(t)b_i^k(t) + \sum_{i=1}^{I} f_i^k(t)h_i^k(t) - y^{k-}(t-1)(1 + \delta^k(t-1)) - \\ - L^k(t) + y^{k-}(t). \end{aligned}$$

Since we do not consider wealth accumulation, the flow balance constraints for assets are $h_i^k(t) = h_i^k(t-1) + b_i^k(t) - s_i^k(t)$. The decision variables $h_i^k(t), b_i^k(t), s_i^k, y^{k-}(t)$ are nonnegative. We can include further constraints forcing a diversification limit investments in risky or illiquid asset classes, limit borrowings etc.

The objective function is mostly related to the wealth at the end of the planning horizon T_0. For each scenario this consists of the amount of the total wealth $\sum_{i=0}^{I} w_i^k(T_0)$ reduced for the present value of liabilities and loans outstanding at the horizon. Risk can also be included in the model in the objective function or in the constraints by choice of suitable utility function or risk measure. To initiate the model, we use scenarios $r_i^k(t), \delta^k(t), f_i^k(t), L^k(t)$ of the returns, interest rates and liabilities for all t and we start with the known, scenario independent initial holdings $h_i^k(0) \equiv h_i(0)$ of cash and all considered assets with $y^{k-} \equiv 0\, \text{for}\, \forall k$. In

case of no ties in scenarios, we can visualize them as a fan of individual scenarios starting from the common known values $r_i(0), \delta(0), f_i(0), L(0)$. Mostly, the input is in a form of a scenario tree, where the nonanticipativity constraints on decisions enter the problem formulation explicitly by using a decision tree following the structure of the already given scenario tree.

Bond Portfolio Management

The aim of the model [14] is to design a decision support for multiperiod management of portfolios of fixed income securities, bonds, in commercial banks. The portfolio manager has an inventory of bonds and cash and at the beginning of each period, he has to decide, which bonds to hold, sell or buy. The structure of the portfolio depends on random cash inflows and outflows, on interest rates, etc. It is supposed the random variables have a discrete probability distribution. We consider T periods and the decisions at the beginning of t^{th}-period ($t = 1, \ldots, T$) depends on the realizations of the random subsequences $\omega^{t-1,\bullet}$. Stages coincide with time periods. The first-stage decisions do not depend on the scenario, the last stage decisions made at the beginning of the T^{th}-period depend on $\omega^{T-1,\bullet}$. The decisions process is assumed to depend of the probability distribution of ω_T, but not by its realization.

The portfolio manager makes decisions in order to maximize the expected market value of the portfolio at the horizon T under constraints on cash-flow, inventory balance, capital losses, initial holdings and nonnegativity of all variables.

We will denote by $b_i(t, \omega^{t-1,\bullet})$ the amount of bond i at the beginning of period t (in dollars of initial purchase price), by $s_i(\tau, t, \omega^{t-1,\bullet})$ the amount of bond i purchased at the beginning of period τ and sold at the beginning of period t (in dollars of initial purchase price), and by $h_i(\tau, t, \omega^{t-1,\bullet})$ the amount of bond i purchased at the beginning of period τ and held at the beginning of period t (in dollars of initial purchase price). We require, that these decision variables are nonnegative and fulfill the constraints on *initial holdings* got before the first period, e.i. $h_i(0,0) = h_i^0$, $i = 1, \ldots, I$, where I is the number of bonds, and on *inventory balance*:

$$\begin{aligned} -h_i(\tau, t-1, \omega^{t-2,\bullet}) + s_i(\tau, t, \omega^{t-1,\bullet}) + h_i(\tau, t, \omega^{t-1}) = 0,\ \tau = 0, 1, \ldots, t-2 \\ -b_i(t-1, \omega^{t-2,\bullet}) + s_i(t-1, t, \omega^{t-1,\bullet}) + h_i(t-1, t, \omega^{t-1,\bullet}) = 0 \\ \text{for all } \omega^{t-1,\bullet}, t = 1, \ldots, T, i = 1, \ldots, I. \end{aligned}$$

For all $\tau = 0, \ldots, t-1$, $t = 1, \ldots, T$, $i = 1, \ldots, I$ and all scenarios we have

$$h_i(\tau, t, \omega^{t-1,\bullet}) = b_i(\tau, \omega^{\tau-1,\bullet}) - \sum_{\nu=\tau+1}^{t} s_i(\tau, \nu, \omega^{\nu-1,\bullet})$$

We will denote by $g_i(\tau, t, \omega^{t-1,\bullet})$ the capital gain/loss on bound i purchased at the beginning of period τ and sold at the beginning of period t (per dollar of initial purchase price, after tax), by $r_i(t, \omega^{t-1,\bullet})$ the annual yield from coupons of bond i bought at the beginning of period t (per dollar of initial purchase price, after tax), and by $w(t, \omega^{t-1,\bullet})$ exogeneous incremental amount of funds at the beginning of period t. The *cash flow constraints* can be written as

$$\sum_{i=1}^{I} b_i(t,\omega^{t-1,\bullet}) - \sum_{i=1}^{I}\sum_{\tau=0}^{t-1}(1+g_i(\tau,t,\omega^{t-1,\bullet}))s_i(\tau,t,\omega^{t-1,\bullet}) -$$
$$-\sum_{i=1}^{I}\left(\sum_{\tau=0}^{t-2} r_i(\tau,\omega^{\tau-1})h_i(\tau,t-1,\omega^{t-2,\bullet}) + r_i(t-1,\omega^{t-2,\bullet})b_i(t-1,\omega^{t-2,\bullet})\right) =$$
$$= w(t,\omega^{t-1,\bullet}) \quad \text{for} \quad \forall\,\omega^{t-1,\bullet}, t=1,\ldots,T.$$

If we want to incorporate the transaction costs, we adjust the gain coefficients for the broker's commission.

We write the constraints on *capital losses* as

$$-\sum_{i=1}^{I}\sum_{\tau=t'}^{t} g_i(\tau,t,\omega^{t-1,\bullet})s_i(\tau,t,\omega^{t-1,\bullet}) \le L(t,\omega^{t-1,\bullet}) \quad \text{for} \quad \forall\,\omega^{t-1,\bullet}, t\in\mathcal{T}',$$

where $L(t,\omega^{t-1,\bullet})$ is the upper bound on the realized capital losses (after tax) from sales during a year, $\mathcal{T}'$ is the set of indices of periods that correspond to the end of fiscal years and t' is the index of the first period in the fiscal year indexed by $t\in\mathcal{T}'$.

The final expected cash value (expectation with respect to ω_T conditioned by $\omega^{T-1,\bullet}$) per dollar of the initial purchase price of the i^{th}-bond purchased at the beginning of period τ and held at the beginning of period T or bought at the beginning of period T is denoted by $\bar{v}_i(\tau,T,\omega^{T-1,\bullet})$. The path probability of the partial sequence $\omega^{T-1,\bullet}$ is denoted by $p(\omega^{T-1,\bullet})$. We write *the objective function* as

$$\sum_{\omega^{T-1,\bullet}} p(\omega^{T-1,\bullet})\sum_{i=1}^{I}[\sum_{\tau=0}^{T-1}(r_i(\tau,\omega^{\tau-1,\bullet}) + \bar{v}_i(\tau,T,\omega^{T-1,\bullet}))h_i(\tau,T,\omega^{T-1,\bullet}) +$$
$$+(r_i(T,\omega^{T-1,\bullet}) + \bar{v}_i(T,T,\omega^{T-1,\bullet}))b_i(T,\omega^{T-1,\bullet})].$$

The aim of the portfolio manager is to *maximize* the objective function subject to constraints on inventory balance, cash flow constraints and constraints on capital losses. It is a large scale linear program and the number of constraints depends on the number of possible partial sequences $\omega^{T-1,\bullet}$, for $t=1,\ldots,T-1$ and of course the more stages we have, the more the number of constraints increases.

Other Applications in Economics and Finance

One common application of multistage stochastic programs is the ALM (asset and liability) model, as in [29]. The aim is to maximize the discounted net value of bank profits minus the expected penalty costs for infeasibility with respect to constraints including legal and policy restrictions, structure of cash flows, liquidity, etc. The source of randomness here are the deposits flows assigning deterministic fixed values to the rates of return and to the interest rates.

In [16], a multi-period portfolio optimization model for portfolio managers in the new fixed-income securities, that have to deal with uncertainty coming not only from the usual interest rate changes, but also from the timing and amount of cash flows, changes in the default and other risk premia and liquidity. In [32], a framework for modeling financial planning problems based on multistage stochastic optimization, e.g. investment strategies (asset allocation strategies),

liability decisions (e.g. borrowings), saving or re-investment decisions, etc., is presented.

Other field of using multistage stochastic models are life insurance and pension funds models, where except for market risks, the mortality is another risk factor (independent of market risks), [39]. It is distinguished between individual contract models, where the random residual lifetime of the contract holder enters the scenario process, and large contract portfolios, where the mortality risk enters as a continuous factor, which determines the total amount of liabilities within every period.

The uncertainties cause various financial risks, such as market risk, currency risk, credit risk, liquidity risk, volatility risk, etc. The risk can be quantified (e.g. variance, VaR). Scenario-based risk management tools are presented in [52], chapter 27.

7.2 Applications in Industry and Other Fields

Production Planning

Production managers often deal with the task to plan the production and utilize it, while coping with demand uncertainty. They should provide production plans over several periods, so that the production can go on continuously. In the paper [15], some simplifications are considered, e.g. the production is always completed in the period in which it begins, at any given period, whatever cannot be produced in-house can be produced at a vendor, all products can be manufactured on the same set of machines and the raw materials are available in the required quantities. The goal is to minimize the total cost, discounted to its net present value, of inventory holding and vendor production. The goal could be also formulated as maximizing the expected difference between total revenue and cost. We consider J products, $j = 1, \ldots, J$, T periods, $t = 1, \ldots, T$, and R manufacturing machines, $r = 1, \ldots, R$. The scenarios (demand outlooks up to the horizon T) are denoted by $k = 1, \ldots, K$. We have *deterministic data* consisting of the inventory holding cost per unit of product j in period t denoted by $h_j(t)$, the unit cost of product j obtained from the vendor in period t denoted by $q_j(t)$, the amount of capacity of machine r needed for producing one unit of product j denoted by a_{jr}, the available capacity of machine r in period t denoted by $K_r(t)$ and the initial inventory of product j denoted by I_{j0}. The demand for product j in period t under scenario k, denoted by $d_j^k(t)$, is uncertain. The decision variables are the inventory volume of product j at the end of period t under scenario k denoted by $I_j^k(t)$, the production volume of product j in period t under scenario k denoted by $x_j^k(t)$ and the amount of product j obtained from the vendor in period t under scenario k denoted by $y_j^k(t)$.

We assume nonanticipativity property, e.g. the decisions are made sequentially using only the past information. The nonanticipativity constraints in the model will be expressed as $\mathbf{x} \in \mathcal{N}, \mathbf{y} \in \mathcal{N}, \mathbf{I} \in \mathcal{N}$. As in [14], the problem can be written

$$\min \sum_{k=1}^{K} p^k \left\{ \sum_{j=1}^{J} \sum_{t=1}^{T} h_j(t) I_j^k(t) + \sum_{j=1}^{J} \sum_{t=1}^{T} q_j(t) y_j^k(t) \right\}$$

subject to

$$
\begin{aligned}
I_j^k(t-1) - I_j^k(t) + x_j^k(t) + y_j^k(t) &= d_j^k(t) && \text{for } \forall j,t,k \\
\sum_{j=1}^{J} a_{rj} x_j^k(t) &\le K_r(t) && \text{for } \forall r,t \\
x_j^k(t) \ge 0, y_j^k(t) \ge 0, I_j^k(t) &\ge 0 && \text{for } j,t,k \\
\mathbf{x} \in \mathcal{N}, \mathbf{y} \in \mathcal{N}, \mathbf{I} &\in \mathcal{N}, &&
\end{aligned}
$$

where p^k is the probability (or a weight) of scenario k.

For the case of no alternative source of production, the amount of product j obtained from the vendor in period t under scenario k, $y_j^k(t)$, is replaced by the lost demand of product j in period t under scenario k, denoted by $b_j^k(t)$ and the unit cost of product j obtained from the vendor in period t, $q_j(t)$, is replaced by the per unit revenue for product j in period t, $l_j(t)$. Sometimes, a penalty function of the unserved demand or a measure of risk can be incorporated in the objective function.

Optimal Transportation

Suppose two places, A, B and a train commuting between them regularly many times per day. One example of that could be the train service providing transport from the airport (A) to the city centre (B). For simplicity, we suppose there are no stations between the airport and the city centre. The problem is the planning of optimal travel service, so that we fulfill the demand requirements - there should be enough trains travelling in one or the other direction within a time interval, so that all people that want to travel in the certain direction within that time interval should have the possibility to do so. At the same time we want to minimize the costs, e.g. the number of trains travelling. Suppose T time intervals, $t = 1, \ldots, T$, with equal length, e.g. the day could be divided into T equally long time intervals. The length of one time interval will be the same as the time needed to travel from A to B (or symmetrically from B to A). At the beginning of each time interval we have to decide, how many trains should depart from A to B and from B to A. For example, if we decide for 3 trains to depart during the next time interval from A to B, we divide the beginning time interval into 3 time intervals with equal length and at the beginning of each subinterval defined in that way, one train departs in the direction $A \to B$. Suppose a time interval $[t_1, t_2]$ divided into 3 subintervals of the same length, $[t_1, i_1), [i_1, i_2), [i_2, t_2]$, the trains depart then at times t_1, i_1 and i_2. We also have to decide at the same time about trains travelling from B to A during the next time interval, we do it analogously as from A to B. At time T (end of the last interval), we want to have the same number of trains in the depot A, as it was at the beginning of the first time interval. The same condition holds for the depot B. We suppose that when a train that travels on the route $A \to B$ arrives at B, it is ready to travel on the route back, $B \to A$.

In the model, we have the following *deterministic* data:

c ... cost of one way of the train in the direction $A \to B$ or $B \to A$;

h ... capacity of one train (maximum number of people that can travel in it at the same time);

b_A ... number of trains in the depot in A at the beginning of the first time

interval;

b_B ... number of trains in the depot in B at the beginning of the first time interval.

The *uncertainty* lies in the number of passengers that want to travel from one point to the other in the time interval. We suppose K scenarios, $k = 1, \ldots, K$. We will denote by

$d_A^k(t)$ ... the number of passengers that want to travel from A to B in the t^{th}-time interval under scenario k;

$d_B^k(t)$ ... the number of passengers that want to travel from B to A in the t^{th}-time interval under scenario k.

The *decision variables* are the number of trains setting out from point A or B within the time interval,

$v_A^k(t)$ ... the number of trains departing from A within the t^{th}-time period under scenario k;

$v_B^k(t)$ ... the number of trains departing from B within the t^{th}-time period under scenario k.

As usual, we assume that at the time of making decisions, we use only the past information. We will summarize the nonaticipativity constraints simply as $\mathbf{v}_A \in \mathcal{N}_A, \mathbf{v}_B \in \mathcal{N}_B$. The problem formulation is then

$$\min \sum_{k=1}^{K} p^k \left\{ \sum_{t=1}^{T} \left(v_A^k(t) + v_B^k(t) \right) \right\}$$

subject to

$$\begin{aligned}
hv_A^k(t) &\geq d_A^k(t) \qquad \text{for} \qquad \forall\, k,t \\
hv_B^k(t) &\geq d_B^k(t) \qquad \text{for} \qquad \forall\, k,t \\
v_A^k(t) &\leq b_A - \sum_{j=1}^{t-1} v_A^k(j) + \sum_{j=1}^{t-1} v_B^k(j) \qquad \text{for} \qquad \forall\, k,t \\
v_B^k(t) &\leq b_B - \sum_{j=1}^{t-1} v_B^k(j) + \sum_{j=1}^{t-1} v_A^k(j) \qquad \text{for} \qquad \forall\, k,t \\
\sum_{t=1}^{T} v_A^k(t) &= \sum_{t=1}^{T} v_B^k(t) \qquad \text{for} \qquad \forall\, k \\
v_A^k(t) \geq 0, v_B^k(t) &\geq 0 \qquad \text{for} \qquad \forall\, k,t \\
\mathbf{v}_A \in \mathcal{N}_A, \mathbf{v}_B &\in \mathcal{N}_B,
\end{aligned}$$

where p^k is the probability of of scenario k.

Rock-Paper-Scissors without Prespecified Number of Rounds

Suppose two players, X, Y, playing the common game *rock-paper-scissors*. At each round of the game, players show simultaneously one of the symbols and the winner is the player, who showed the stronger symbol in the combination. We will denote the symbols by **R** - rock, **P** - paper, **S** - scissors. The symbol with star in the combination is the winning symbol in the particular combination, $\mathbf{R}^* \leftrightarrow \mathbf{S}, \mathbf{S}^* \leftrightarrow \mathbf{P}, \mathbf{P}^* \leftrightarrow \mathbf{R}$. Suppose, that the number of rounds is not specified in advance and it depends on the occurence of some external event, e.g. small children playing the in late the evening and they must stop when their mother comes and tells them off. We will denote by r the total number of rounds and for the game we will consider the the expected value of the total number of

rounds $T = \mathbf{E}[r]$. We will denote the set of possible symbols of each player as $\mathbf{y} = (\mathbf{R}, \mathbf{P}, \mathbf{S})$ and $\mathbf{x} = (\mathbf{R}, \mathbf{P}, \mathbf{S})$, respectively. Each player has the same amount of coins W at the beginning of the game. If player X wins, he gets one coin from player Y, if he loses, he gives player Y one coin. In case of a draw, players do not pay anything. Suppose, we play as the player X. We will denote the function, which determines if player X gets or gives one coin in round $t, t = 1, \ldots, T$, by $F_t(x_t, y_t)$, where x_t is the symbol showed by player X in the t^{th}-round and y_t is the symbol showed by player Y in the t^{th}-round. In case of our win as player X, the function is $F_t(x_t, y_t) = 1$, (x_t is the winning symbol in the combination (x_t, y_t) in round t and we get one coin from player Y), on the contrary, in case of our loss, it is $F_t(x_t, y_t) = -1$,(player Y showed the winning symbol in the combination in round t), in case of a draw $F_t(x_t, y_t) = 0$, (the same symbols showed by both players in the combination in round t). As player X, in each round we have to decide which symbol to show, so that we win. Since we show our chosen symbol at the same time when player Y shows his symbol, in round t before showing symbols, we know only the history from rounds $1, \ldots, t-1$ (nonanticipativity conditions). The random data is the symbol y_t showed by the opponent Y in each round, our decision of the symbol showed at stage t is x_t. Suppose a scenario tree of the symbols showed by player Y in each round (e.g. Figure F.7.1.).

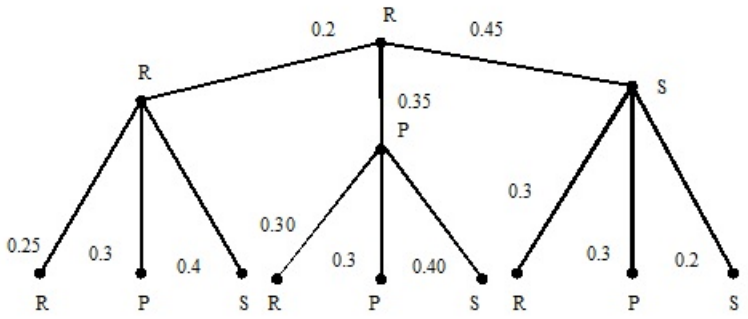


Figure F.7.1. Scenario tree for the symbols of player Y.

Suppose K different scenarios, with respective probabilities p^k. Our aim as player X is to maximize our profit at the end of the game, e.g. have as much coins as possible after round T. At the beginning of each round, the number of coins of each player must be positive. The nonanticipativity constraints are written as $\mathbf{x}, \mathbf{y} \in \mathcal{N}$. As player X, the task for us is

$$\max \sum_{k=1}^{K} p^k \sum_{t=1}^{T} F_t(x_t^k, y_t^k)$$

subject to

$$W \geq \max \left\{ \sum_{i=1}^{t-1} F_i(x_t^k, y_t^k)\,;\, -\sum_{i=1}^{t-1} F_i(x_t^k, y_t^k) \right\} \qquad \text{for} \qquad \forall t, k$$
$$\mathbf{x}, \mathbf{y} \in \mathcal{N}.$$

Other Examples of Applications in Industry and Other Fields

An interesting application is for electric power generation systems. They should serve the uncertain time varying demand for electricity (*load*). We need to determine the numbers or capacities of R different equipments or technologies characterized by unit investment costs and unit operating costs to serve the demand in such a way, that the expected total cost for capacity expansion is minimal. The data for the analysis of this problem are usually based on extensive data sets and historical time series. We should also consider the time evolution of costs and of the load duration curve, appearance of new technologies, construction delays, etc. The random parameters include, except for the random demand, also the costs, random construction delays, random lifetime of equipments, etc. An example of such model over T periods with decision variables and model parameters dependent on the index of the period and on the considered random factors, demands and costs is shown in [14].

In the field of production, melt control is studied as one of the production steps in iron and steel works, see [52], chapter 15 (Melt Control : Charge Optimization via Stochastic Programming by J. Dupačová and P.Popela). Produced alloys and input materials are composed of basic elements(iron, carbon, etc.) and the production process consists of several steps (e.g. charge, alloying). The hot melt in the furnace is enriched with input materials (return materials, scrap, ferroalloys, etc.) during the alloying process. and the hot mixture is melted again. The problem has a multistage decision structure. The unit costs of the inputs are known at the time of decision making, but the composition of input materials is not known precisely. The melt composition changes in each step of the process and also random losses of elements in the melt must be taken into consideration. The amounts of elements change randomly during heating of the melt (e.g. due to a rise of slag and oxidation). The losses are dependent on the composition of the melted materials and they might be influenced also by the amounts of these materials used. We express the remaining amount of an element as a linear function in the input quantities of all considered elements and we call the coefficients utilizations of the considered element related to the amount of other elements in the melt. The task is finding amounts of the input materials in the lowest cost in order to achieve the prescribed output alloy composition. Scenario-based two- and three-stage stochastic linear programs illustrating these modeling ideas are also presented in the given chapter.

The problem of an optimal scheduling of the generating capacity among generating units of an existing power system is described in [14]. The aim of the schedule is to minimize the generation costs and meet the demand and other constraints imposed by the physical characteristic of the system. At each period, we need to decide which units to commit and at what generating capacity. The uncertainty lies in the demand and the spot market prices at each time period.

7.3 Example of Usage of SCENRED 2

For scenario generation and reduction, one can use GAMS\SCENRED 2 and the model *SRTREE* introduced by Heistch, H., Römisch, W. and Strugarek, C. in GAMS. First we had scenario data and a fan of scenarios was built. Then we used backward and forward tree construction ([20]) to build a tree. The backward reduction and the forward selection method were used on the fan, both methods resulted in the same reduced tree. Suppose 5 different scenarios and 6 stages. For each stage and scenario, we have 5 random data values, r_1, r_2, r_3, r_4, r_5. The scenario data are in the form of the table below, e.g. in the row $s_k.tj$ and column r_i we find the realization of r_i in scenario s_k at stage t_j. We assign to each scenario the following probabilities: $s1 \to 0.2, s2 \to 0.1, s3 \to 0.3, s4 \to 0.2, s5 \to 0.2$.The following data were used:

	r1	r2	r3	r4	r5
s1.t1	42.5	9.1	7.5	120.0	145.0
s1.t2	39.8	11.5	8.4	90.0	112.0
s1.t3	37.6	14.4	6.7	109.1	110.0
s1.t4	38.8	14.7	8.9	134.0	141.0
s1.t5	40.3	14.1	7.2	135.0	200.0
s1.t6	45.5	15.1	9.1	131.9	199.0
s2.t1	42.5	9.1	7.5	120.0	130.2
s2.t2	41.8	11.2	8.5	93.5	120.1
s2.t3	40.6	15.7	7.8	107.0	121.9
s2.t4	39.9	11.4	8.5	130.2	131.8
s2.t5	38.4	15.6	9.3	101.0	132.1
s2.t6	41.3	16.1	9.1	101.0	132.1
s3.t1	42.5	9.1	7.5	120.0	156.1
s3.t2	35.8	11.7	9.0	100.0	112.3
s3.t3	37.6	14.0	6.3	110.0	111.3
s3.t4	35.7	13.9	7.5	118.0	119.1
s3.t5	38.6	15.9	9.3	81.1	82.1
s3.t6	39.1	16.1	9.4	90.2	94.1
s4.t1	42.5	9.1	7.5	120.0	145.1
s4.t2	39.8	11.2	11.5	91.0	110.1
s4.t3	38.7	17.0	6.9	112.0	115.6
s4.t4	33.8	16.1	7.9	125.0	125.0
s4.t5	36.3	13.7	11.3	92.2	93.1
s4.t6	36.1	13.8	12.3	91.3	100.3
s5.t1	42.5	9.3	7.6	121.1	145.2
s5.t2	39.2	11.5	11.5	91.3	110.1
s5.t3	38.7	17.7	6.4	112.1	115.6
s5.t4	33.7	17.1	7.1	125.0	125.4
s5.t5	36.5	13.6	11.3	92.1	93.1
s5.t6	36.2	13.8	12.5	91.3	100.1

We identify the nodes by a sequence of integer numbers, where we naturally assign number 1 to the root node. In the following results from SCENRED, we will have a canonical order by stages and scenarios. The predecessor of the root node is set as 1. First, a fan of scenarios was generated (Figure F.7.3.1). The figure shows the fan with numbers assigned to the nodes, the ordering in the picture goes from up to down for each stage.

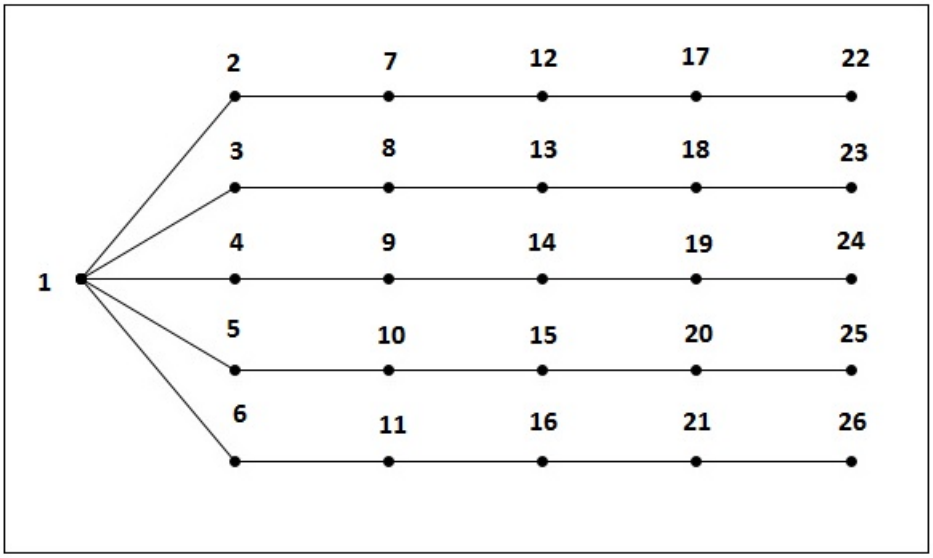


Figure F.7.3.1. A fan of scenarios with numbers assigned to each node.

In the following table, we have the nodes denoted by the number of their predecessor and the probabilities are the conditional probabilities (the probability of moving from the predecessor to that particular node). For each node, we have the values of the random data. In the fan of scenarios, we have 26 nodes.

```
NODES    26
RANDOM   5

DATA
 PRED   PROB     R1        R2        R3        R4        R5
 1      1.00    42.50      9.10      7.50    120.00    145.00
 1      0.20    39.20     11.50     11.50     91.30    110.10
 1      0.20    39.80     11.20     11.50     91.00    110.10
 1      0.30    35.80     11.70      9.00    100.00    112.30
 1      0.10    41.80     11.20      8.50     93.50    120.10
 1      0.20    39.80     11.50      8.40     90.00    112.00
 2      1.00    38.70     17.70      6.40    112.10    115.60
 3      1.00    38.70     17.00      6.90    112.00    115.60
 4      1.00    37.60     14.00      6.30    110.00    111.30
 5      1.00    40.60     15.70      7.80    107.00    121.90
 6      1.00    37.60     14.40      6.70    109.10    110.00
```

```
 7      1.00    33.70    17.10     7.10    125.10    125.40
 8      1.00    33.80    16.10     7.90    125.00    125.00
 9      1.00    35.70    13.90     7.50    118.00    119.10
 10     1.00    39.90    11.40     8.50    130.20    131.80
 11     1.00    38.80    14.70     8.90    134.00    141.00
 12     1.00    36.50    13.60    11.30     92.10     93.10
 13     1.00    36.30    13.80    12.30     91.30    100.30
 14     1.00    36.60    15.90     9.30     81.10     82.10
 15     1.00    38.40    15.60     9.30    101.00    132.10
 16     1.00    40.30    14.10     7.20    135.00    200.00
 17     1.00    36.20    13.80    12.50     91.30    100.10
 18     1.00    36.10    13.80    12.30     91.30    100.30
 19     1.00    39.10    16.10     9.40     90.20     94.10
 20     1.00    41.30    16.10     9.10    101.00    132.10
 21     1.00    45.50    15.10     9.10    131.90    199.00
END
```

Then the **backward construction method** was used giving us the following results. For the predecessor relationship, in the table with numerical results, we have two nodes with predecessor 4, the first node in the table with predecessor 4 is in fact node number 6 and the second node with predecessor 4 is in fact node with number 7. This means, that if we have for one predecessor several descendants, the table keeps the orders, i.e. the first node with that predecessor in the table is the descendant of that predecessor having the lowest number among all the descendants, the second descendant has the second-lowest number among the descendants of this particular predecessor, etc.

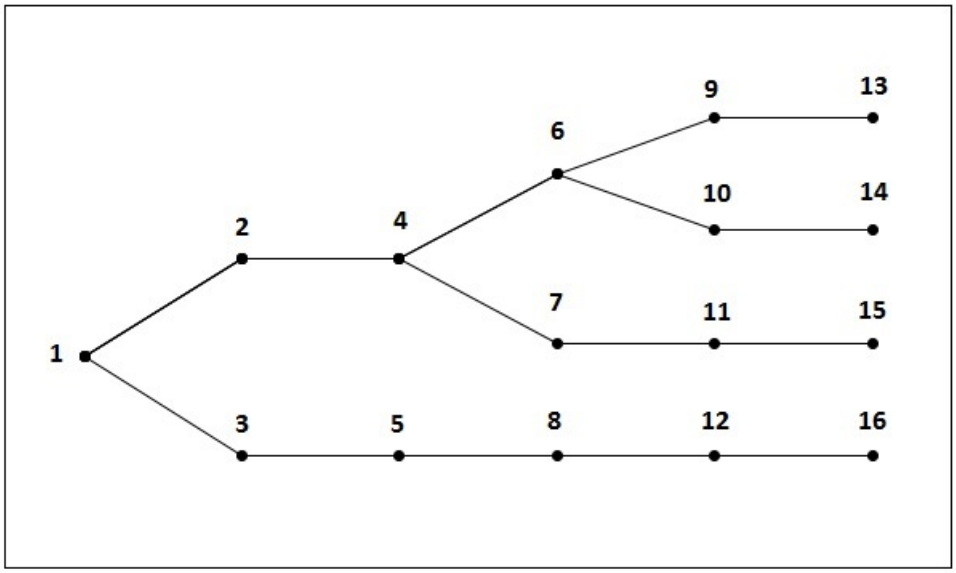


Figure F.7.3.2 A tree obtained from the backward tree construction.

The numerical results are as follows:

```
NODES    16
RANDOM   5
```

```
DATA
 PRED    PROB     R1       R2       R3       R4       R5
 1       1.00    42.50     9.10     7.50   120.00   145.00
 1       0.60    39.80    11.50     8.40    90.00   112.00
 1       0.40    39.20    11.50    11.50    91.30   110.10
 2       0.60    37.60    14.40     6.70   109.10   110.00
 3       0.40    38.70    17.70     6.40   112.10   115.60
 4       0.30    38.80    14.70     8.90   134.00   141.00
 4       0.30    35.70    13.90     7.50   118.00   119.10
 5       0.40    33.70    17.10     7.10   125.00   125.40
 6       0.20    40.30    14.10     7.20   135.00   200.00
 6       0.10    38.40    15.60     9.30   101.00   132.10
 7       0.30    38.60    15.90     9.30    81.10    82.10
 8       0.40    36.50    13.60    11.30    92.10    93.10
 9       0.20    45.50    15.10     9.10   131.90   199.00
 10      0.10    41.30    16.10     9.10   101.00   132.10
 11      0.30    39.10    16.10     9.40    90.20    94.10
 12      0.40    36.20    13.80    12.50    91.30   100.10
END
```

For the results of the **forward tree construction method**, the same rules for interpretation of the table of results hold. We have more (20) nodes than that we obtained in the backward tree construction (see the table on the next page). The scenario tree from forward tree construction is depicted in Figure F.7.3.3.

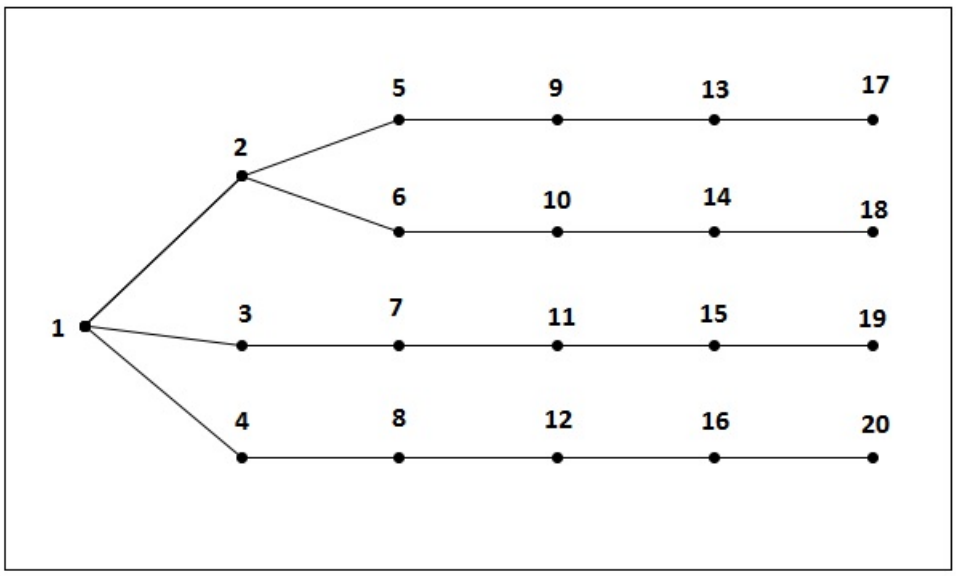


Figure F.7.3.3 A tree obtained from the forward tree construction.

```
NODES   20
RANDOM  5

DATA
 PRED  PROB   R1      R2      R3       R4       R5
 1     1.00   42.50    9.10    7.50   120.00   145.00
 1     0.30   39.80   11.50    8.40    90.00   112.00
 1     0.30   35.80   11.70    9.00   100.00   112.30
 1     0.40   39.80   11.20   11.50    91.00   110.10
 2     0.20   37.60   14.40    6.70   109.10   110.00
 2     0.10   40.60   15.70    7.80   107.00   121.90
 3     0.30   37.60   14.00    6.30   110.00   111.30
 4     0.40   38.70   17.00    6.90   112.00   115.60
 5     0.20   38.80   14.70    8.90   134.00   141.00
 6     0.10   39.90   11.40    8.50   130.20   131.80
 7     0.30   35.70   13.90    7.50   118.00   119.10
 8     0.40   33.80   16.10    7.90   125.00   125.00
 9     0.20   40.30   14.10    7.20   135.00   200.00
 10    0.10   38.40   15.60    9.30   101.00   132.10
 11    0.30   38.60   15.90    9.30    81.10    82.10
 12    0.40   36.30   13.70   11.30    92.20    93.10
 13    0.20   45.50   15.10    9.10   131.90   199.00
 14    0.10   41.30   16.10    9.10   101.00   132.10
 15    0.30   39.10   16.10    9.40    90.20    94.10
 16    0.40   36.10   13.80   12.30    91.30   100.30
END
```

As we can see in the Appendix, when the backward scenario reduction and forward scenario selection method were used on the fan of scenario that we had, both methods gave us the same structure of the tree with the same probabilities and the same scenarios. Also both these methods gave the same tree structure, probabilities and scenarios as we obtained from the backward scenario tree construction.

Epilogue

Starting with some basic theory and continuing with two-stage problems, we arrived at the concept of multistage stochastic programs and their different forms (linear multistage stochastic programs, convex multistage programs). The main focus was on formulating multistage stochastic programs for the special case of scenario approach. Since the good approximation of the true probability distribution (in terms of difference in the optimal value) is very important, several methods of scenario generation were introduced. The scenario tree reduction methods and some related methods were briefly described. The importance of the concept of multistage stochastic programs theory was illustrated by many examples from the field of finance, industry, transportation planning and games. If we think of approximating these real problems, scenarios seem to be a quite natural choice. The idea of the random number of stages was sketched in the *Rock-Paper-Scissors* example, where the expected value of the number of stages was considered. Going more into detail, it could be interesting to study problems with random number of stages, where the number of stages does not depend on the underlying stochastic data process, which accounts for the randomness in the program (e.g. in the *Rock-Paper-Scissors* example, we assumed, that the number of stages (till the time, when the mother comes) was independent of the strategy of our opponent (the random process)). On the other hand, studying problems, where the random number of stages depends on the random process could be practical for modeling some specific decision problems (e.g. the investor invests till the random interest rate is within a certain interval).

Bibliography

[1] Batlle, C. and Barquìn, J. (2004). Fuel prices scenario generation based on a multivariate GARCH model for risk analysis in a wholesale electricity market. *Electrical Power and Energy Systems 36*, 273-280, ELSEVIER

[2] Beltratti, A., Consiglio, A. and Zenios, S.A. (1999). Scenario modeling for the management of international bond portfolios. *Annals of Operations Research* 85, 227–247.

[3] Black, F., Derman, E. and Toy, W. (1990). A One Factor Model of Interest Rates and Its Application to Treasury Bond Options. *Financial Analysts Journal*, January-February 1990, 33-39

[4] Cario, M.C. and Nelson, B.L. (1997). Modeling and Generating Random Vectors with Arbitrary Marginal Distributions and Correlation Matrix. *Technical Report, Department of Industrial Engineering and Management Sciences*, Northwestern University, Evaston, Illinois

[5] Clemen, R.T. and Reilly, T. (1999). Correlations and Copulas for Decision and Risk Analysis. *Management Science, 45(2)*, p.208-224

[6] Consigli, G. and Dempster, M. A. H. (1997). Dynamic Stochastic Programming for Asset-Liability Management. *Baltzer Journals*

[7] Deler, B. and Nelson, B. L. (2000). Modeling and Generating Multivariate Time Series with Arbitrary Marginals Using an Autoregressive Technique. *Technical Report, Department of Industrial Engineering and Management Sciences*, Northwestern University, Evaston, Illinois

[8] Dempster, M.A.H. and Thompson, R.T. (1999). EVPI-based Importance Sampling Solution Procedures for Multistage Stochastic Linear Programmes on PArallel MIMD architectures. *Annals of Operations Research, 90*, p. 161-184

[9] Dupačová, J. (2006). *Contamination for Multistage Stochastic Programs*, Prague Stochastics 2006

[10] Dupačová, J. (2009). *Portfolio Optimization and Risk Management via Stochastic Programming*, Osaka University Press, Osaka, Japan

[11] Dupačová, J. (1998). Stochastic Programming : Approximation via scenarios. *Aportaciones Matemáticas, Serie Communicationes*, 77-94, Sociedad Matemática Mexicana

[12] Dupačová, J., Consigli, G. and Wallace, S. W. (2000). Scenarios for Multistage Stochastic Programs. *Annals of Operations Research* 100, 25–53, Kluwer Academic Publishers

[13] Dupačová, J., Gröwe-Kuska, N. and Römisch, W. (2003). Scenario Reducton In Stochastic Programming : An Approach Using Probability Metrics. *Mathematical Programming, Series A 95,* p. 493-511

[14] Dupačová, J., Hurt, J. and Štěpán, J. (2002). *Stochastic Modeling in Economics and Finance*, Kluwer Academic Publishers, Dordrecht

[15] Escudero, L. F. et. al., (1993). Production Planning via Scenario Modeling. *Annals of Operations Research 43*, p. 311-335

[16] Golub, B., Holmer, M., McKendall, R., Pohlman, L. and Zenios, S. A. (1995). A Stochastic Programming Model for Money Management. *European Journal of Operational Research, 85*, p. 282-296

[17] Gröwe-Kuska, N., Heitsch, H. and Römisch, W. (2003). Scenario Reduction and Scenario Tree Construction for Power Management Problems. *Paper for Presentation at 2003 IEEE Bologna PowerTech Conference*

[18] Gülpinar, N., Rustem, B. and Settergren, R. (2004). Simulation and Optimization Approaches to Scenario Tree Generation. *Journal of Economic Dynamics and Control, 28*, p. 1291-1315

[19] Heitsch, H. and Römisch, W. (2011). Scenario Tree Generation for Multistage Stochastic Programs. *Handbook in Operations Research and Management Science, Vol.1*, 313–342.

[20] Heitsch, H. and Römisch, W. (2007). Scenario Tree Modeling for Multistage Stochastic Programs. *Math. Program.* Series A.

[21] Heitsch, H. and Römisch, W. (2009). Scenario Tree Reduction for Multistage Stochastic Programs. *Comput. Manag. Sci.* 6, 117–133

[22] Hochreiter, R. and Pflug, G. Ch. (2007). Financial Scenario Generation for Stochastic Multi-stage Decision Processes as Facility Location Problems *Ann. Oper. Res.* 152, 257–272

[23] Høyland, K., Kaut, M. and Wallace, S.W. (2003). A Heuristic for Moment-Matching Scenario Generation. *Computational Optimization and Applications, 24,* p. 169-185

[24] Høyland, K. and Wallace, S.W. (2001). Generating Scenario Trees for Multistage Decision Problems. *Management Science, 47(2),* p.295-307

[25] Kaut, M. and Wallace, S.W. (2003). Evaluation of Scenario Generation Methods for Stochastic Programming. *Available at:*
http://edoc.hu-berlin.de/series/speps/2003-14/PDF/14.pdf

[26] Kaut, M. and Wallace, S.W. (2011). Shape-based Scenario Generation Using Copulas. *Computational Management Science, 8*, p.181-199

[27] Kouwenberg, R. (2001). Scenario Generation and Stochastic Programming Models for Asset Liability Management *European Journal of Operational Research* 134, 279–292

[28] Küchler, Ch. and Vigerske, S. (2007). Decomposition of Multistage Stochastic Programs with Recombining Scenario Trees. *Preprint, Humboldt-University*

[29] Kusy, M. I. and Ziemba, W. T. (1986). A bank asset and liability management model. *Operations Research, Vol. 34*, p. 356-376

[30] Loretan, M. (1997). Generating Market Risk Scenarios Using Principal Components Analysis : Methodological and Practical Considerations. In *The Measurement of Aggregate Market Risk, CGFS Publications No. 7*, p. 23-60, Bank for International Settlements

[31] Lyhagen, J. (2001). A Method to Generate Multivariate Data with Moments Arbitrary Close to the Desired Moments. *SSE/EFI Working Paper Series in Economics and Finance, No. 481*

[32] Mulvey, J.M. and Shetty, B. (2004). Financial Planning via Multi-stage Stochastic Optimization. *Computers & Operations Research 31,* p. 1-20

[33] Owen, A.B. (2008). Local Antithetic Sampling With Scrambled Nets. *The Annals of Statistics, Vol. 36, No. 5*, p. 2319-2343

[34] Pennanen, T. and Koivu, M. (2002). Integration Quadratures in discretization of Stochastic Programs. *Stochastic Programming E-Print Series*

[35] Pflug, G. Ch. (2001). Scenario Tree Generation for Multiperiod Financial Optimization by Optimal Discretization. *Math. Program.* Series B89, 251–271.

[36] Pflug, G. Ch. (2010). Version-independence and Nested Distributions in Multistage Stochastic Optimization. *SIAM J. Optim.*, 20(3), 1406-1420

[37] Pflug, G. and Pichler, A. (2012). A Distance for Multi-stage Stochastic Optimization Models. *SIAM J.Optim.* 22, 1-23

[38] Pflug, G. and Pichler, A. (2011). Approximations for probability distributions and stochastic optimization problems. In: Consigli, G., Dempster, M. and Bertocchi, M. (eds) *Springer Handbook on Stochastic Optimization Methods in Finance and Energy. International Series in Operational Research and Management Science 163*, chapter 15, 343-387

[39] Pflug, G. Ch. and Römisch, W. (2007). *Modeling, Measuring and Managing Risk*, World Scientific, Singapore.

[40] Rachev, S.T. (1991). *Probability Metrics and The Stability of Stochastic Models* J. Wiley and Sons, New York

[41] Römisch, W. (2009). Scenario Reduction Techniques in Stochastic Programming. In *Stochastic Algorithms : Foundations and Applications, 5th International Symposium, SAGA 2009*, available at: http://www.mathematik.huberlin.de/ romisch/papers/roemischsaga.pdf

[42] Ruszczynski, A. and Shapiro, A. (2003). Stochastic Programming Models. *Handbooks in Oper and MS, Vol.10*, Elsevier Science.

[43] SCENRED 2 manual, available at:
http://www.gams.com/dd/docs/solvers/scenred2.pdf

[44] Shapiro, A. (2003). Monte Carlo Sampling Methods. *Handbooks in Oper and MS, Vol.10*

[45] Shapiro, A., Dentcheva, D. and Ruszczynski, A. (2009). *Lectures on Stochoastic Programming, Modeling and Theory*, United States of America

[46] Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris 8*, 229-231

[47] Sklar, A. (1996). Random Variables, Distribution Functions and Copulas - a Personal Look Backward and Forward. In *Rüschendorff L. Schweizer B, Taylor M (eds) Distributions with Fixed Marginals and Related Topics.* Institute of Mathematical Statistics, Hayward, pp.1-14

[48] Smith, J. E. (1993). Moment Methods for Decision Analysis. *Management Science, 39(3),*, p. 340-358

[49] Timonina, A.V. (2013). Multistage Stochastic Optimization: The Distance between Stochastic Scenario Processes. *Computational Management Science.*, Springer-Verlag Berlin Heidelberg

[50] Vallander, S.S. (1973). Calculation of the Wasserstein Distance between Probability Distributions on the Line. *Teor. Veroyatnost. i Primenen, 18*, 824-827

[51] Villani, C. (2003). Topics in Optimal Transportation. *Graduate Studies in Mathematics, vol. 58.* American Mathematical Society, Providence

[52] Wallace, S. W. and Ziemba, W. T. (2005). *Applications of Stochastic Programming.*, Society for Industrial Mathematics (MPS - SIAM Series in Optimization), Philadelphia

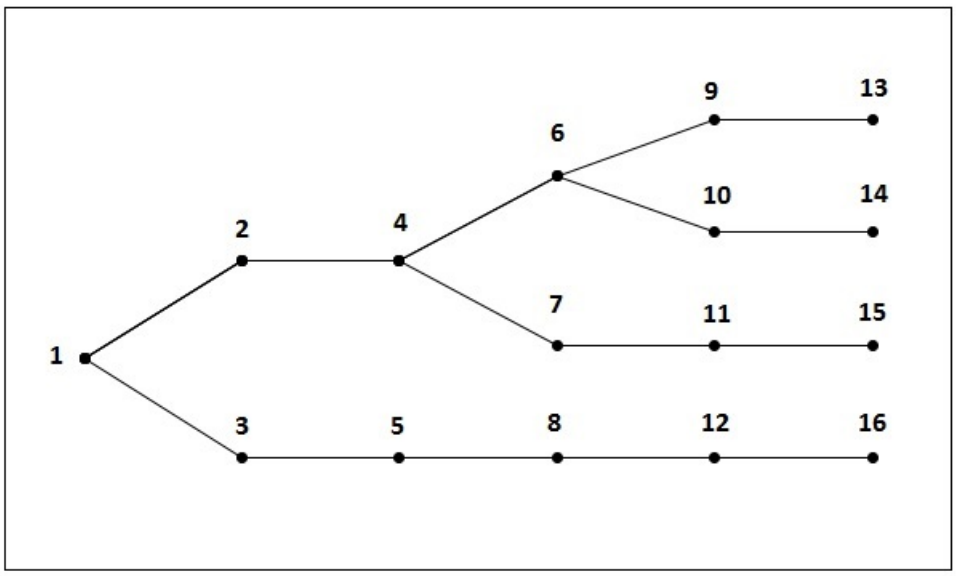
[53] Wets, R. J-B (1989). Stochastic Programming. *Handbook for Operations Research and Management Science, Vol. 1* 573–629

[54] Wets, R. J-B (1989). The Aggregation Principle in Scenario Analysis and Stochastic Optimization. In *Algorithms and Modelling in Mathematical Programming*, Springer-Verlag, Berlin, p. 91-113

[55] Zhou, Q. , Tesfatsion, L. and Liu, Ch. (2009). Scenario Generation for Price Forecasting in Restructured Wholesale Power Markets. *IEEE*

Appendix

The backward reduction method and forward selection method in GAMS \ SCENRED 2 gave us the same structure of the tree with the same probabilities and scenarios, depicted as



Using **backward reduction** an **forward selection**, we obtained identical results:

```
TYPE    TREE

NODES   16
RANDOM  5

DATA
 PRED    PROB     R1       R2       R3       R4       R5
 1       1.00    42.50     9.10     7.50   120.00   145.00
 1       0.60    39.80    11.50     8.40    90.00   112.00
 1       0.40    39.20    11.50    11.50    91.30   110.10
 2       0.60    37.60    14.40     6.70   109.10   110.00
 3       0.40    38.70    17.70     6.40   112.10   115.60
 4       0.30    38.80    14.70     8.90   134.00   141.00
 4       0.30    35.70    13.90     7.50   118.00   119.10
 5       0.40    33.70    17.10     7.10   125.00   125.40
 6       0.20    40.30    14.10     7.20   135.00   200.00
 6       0.10    38.40    15.60     9.30   101.00   132.10
 7       0.30    38.60    15.90     9.30    81.10    82.10
 8       0.40    36.50    13.60    11.30    92.10    93.10
 9       0.20    45.50    15.10     9.10   131.90   199.00
 10      0.10    41.30    16.10     9.10   101.00   132.10
 11      0.30    39.10    16.10     9.40    90.20    94.10
 12      0.40    36.20    13.80    12.50    91.30   100.10
END
```

The code in GAMS \ SCENRED 2 introduced in the model "SRTREE"and modified for scenario construction and reduction, that was used, is:

```
Set s scenarios   / s1*s5 /
    t stage       / t1*t6 /
    r random data / r1*r5 /;

Parameter
    p(s) probability / s1 0.2, s2 0.1, s3 0.3, s4 0.2, s5 0.2 /

Table sdata(s,t,r)
          r1     r2    r3     r4     r5
s1.t1    42.5   9.1   7.5   120.0   145.0
s1.t2    39.8  11.5   8.4    90.0   112.0
s1.t3    37.6  14.4   6.7   109.1   110.0
s1.t4    38.8  14.7   8.9   134.0   141.0
s1.t5    40.3  14.1   7.2   135.0   200.0
s1.t6    45.5  15.1   9.1   131.9   199.0

s2.t1    42.5   9.1   7.5   120.0   130.2
s2.t2    41.8  11.2   8.5    93.5   120.1
s2.t3    40.6  15.7   7.8   107.0   121.9
s2.t4    39.9  11.4   8.5   130.2   131.8
s2.t5    38.4  15.6   9.3   101.0   132.1
s2.t6    41.3  16.1   9.1   101.0   132.1

s3.t1    42.5   9.1   7.5   120.0   156.1
s3.t2    35.8  11.7   9.0   100.0   112.3
s3.t3    37.6  14.0   6.3   110.0   111.3
s3.t4    35.7  13.9   7.5   118.0   119.1
s3.t5    38.6  15.9   9.3    81.1    82.1
s3.t6    39.1  16.1   9.4    90.2    94.1

s4.t1    42.5   9.1   7.5   120.0   145.1
s4.t2    39.8  11.2  11.5    91.0   110.1
s4.t3    38.7  17.0   6.9   112.0   115.6
s4.t4    33.8  16.1   7.9   125.0   125.0
s4.t5    36.3  13.7  11.3    92.2    93.1
s4.t6    36.1  13.8  12.3    91.3   100.3

s5.t1    42.5   9.3   7.6   121.1   145.2
s5.t2    39.2  11.5  11.5    91.3   110.1
s5.t3    38.7  17.7   6.4   112.1   115.6
s5.t4    33.7  17.1   7.1   125.0   125.4
s5.t5    36.5  13.6  11.3    92.1    93.1
s5.t6    36.2  13.8  12.5    91.3   100.1
```

```
;


;
* Construct a fan
$eval nnodes card(s)*(card(t)-1)
set n        nodes / n0*n%nnodes% /
    nn(n)    next nodes
    anc(n,n) ancestor relation;
parameter
    prob(n)  node probability
    rv(r,n)  random values;

nn('n0') = yes;
loop(s,
  loop(t$(ord(t)<card(t)),
     loop(nn(n),
        if (sameas('t1',t),
           anc(n+1,'n0') = yes;
        else
           anc(n+1,n) = yes;
        );
        prob(n+1)=p(s);
        rv(r,n+1) = sdata(s,t+1,r);
     );
     nn(n) = nn(n-1);
  )
);
prob('n0') = 1; rv(r,'n0') = sdata('s1','t1',r);

* Initialize ScenRed2
$set sr2prefix test
$libinclude scenred2

set anc_noloss(n,n), anc_red(n,n);
parameter
    prob_noloss(n),  prob_red(n);

* Scenred2 call, no reduction
ScenredParms('red_percentage') = 0;
ScenredParms('out_tree'    ) = 1;
ScenredParms('visual_init'   ) = 1;


$libinclude runscenred2
%sr2prefix% tree_con n anc prob anc_noloss
prob_noloss rv
abort$(card(anc_noloss)<>15)
```

```
'scenred tree construction gave incorrect noloss
 tree', anc_noloss;
display anc_noloss;

* Scenred2 call backward reduction
ScenredParms('reduction_method') = 2;
ScenredParms('visual_init'   ) = 1;
ScenredParms('visual_red'    ) = 1;
ScenredParms('out_tree'    ) = 1;
$libinclude runscenred2
%sr2prefix%
tree_con n anc prob anc_red prob_red rv
display anc_red;

* Scenred2 call forward reduction
ScenredParms('reduction_method') = 1;
ScenredParms('visual_init'   ) = 1;
ScenredParms('visual_red'    ) = 1;
ScenredParms('out_tree'    ) = 1;
$libinclude runscenred2
%sr2prefix%
tree_con n anc prob anc_red prob_red rv
display anc_red;

* Scenred3 call backward construction
ScenredParms('construction_method') = 2;
ScenredParms('red_percentage') = 0.2;
ScenredParms('visual_init'   ) = 1;
ScenredParms('visual_red'    ) = 1;
ScenredParms('out_tree'    ) = 1;
$libinclude runscenred2
%sr2prefix%
tree_con n anc prob anc_red prob_red rv
display anc_red;

* Scenred4 call forward construction
ScenredParms('construction_method') = 1;
ScenredParms('red_percentage') = 0.2;
ScenredParms('visual_init'   ) = 1;
ScenredParms('visual_red'    ) = 1;
ScenredParms('out_tree'    ) = 1;
$libinclude runscenred2
%sr2prefix%
tree_con n anc prob anc_red prob_red rv
display anc_red;
```